

**DESARROLLO DE UN KIT ROBÓTICO TIPO “SCARA” PARA LA EDUCACIÓN
EN REGIONES**

**ANDERSON ESTEBAN CORREA ARANGO
MARILUZ GARCÍA VERA
ANDERSON LEÓN GIRALDO**

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO
FACULTAD DE INGENIERÍA
TECNOLOGÍA EN SISTEMAS MECATRÓNICOS
MEDELLÍN**

2024

**DESARROLLO DE UN KIT ROBÓTICO TIPO “SCARA” PARA LA EDUCACIÓN
EN REGIONES**

ANDERSON ESTEBAN CORREA ARANGO

MARILUZ GARCÍA VERA

ANDERSON LEÓN GIRALDO

Asesor metodológico

Sergio Hernando Ruiz Obando

Magíster en Tecnologías Digitales Aplicadas a la Educación

Asesor Técnico

Juan Camilo Herrera Pineda

Magíster en Ingeniería Mecánica

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO

FACULTAD DE INGENIERÍA

TECNOLOGÍA EN SISTEMAS MECATRÓNICOS

MEDELLÍN

2024

Contenido

	Pág.
Introducción	13
1. Planteamiento del problema	14
1.1 Descripción	14
1.2 Formulación	15
2. Justificación	16
3. Objetivos.....	18
3.1 Objetivo general	18
3.2 Objetivos específicos	18
4. Marco teórico.....	19
4.1 Estado del arte	19
4.2 Referentes teóricos	22
4.2.1 Términos importantes.	23
4.2.2 Tipos de articulaciones.	27
4.2.3 Clasificación de manipuladores robóticos..	27
4.2.4 Componentes eléctricos y electrónicos.	30
4.2.5 Componentes mecánicos.....	38
5. Metodología.....	43
5.1 Tipo de proyecto	43
5.2 Método.....	44
5.2.1 Diseños Mecánicos, Eléctricos y Electrónicos del Prototipo del Brazo Robótico.. .	45
5.2.2 Desarrollo de los Sistemas Mecánicos, Eléctricos y Electrónicos del Prototipo de un Brazo Robótico..	46
5.2.3 Pruebas de Funcionamiento al Prototipo de Brazo Robótico..	47
5.3 Instrumentos de recolección de información.....	48
5.3.1 Fuentes primarias.	48
5.3.2 Fuentes secundarias.	48
6. Resultados.....	49
6.1 Diseño del prototipo de brazo robótico “SCARA”	49
6.1.1 Modelación de las piezas en CAD.....	49

6.1.2	Diseño de las correas..	66
6.1.3	Simulación y validación del circuito electrónico en Tinkercad.....	67
6.2	Desarrollo del prototipo de brazo robótico “SCARA”	68
6.2.1	Impresión 3D de piezas y correas.	68
6.2.2.	Ensamble físico de piezas.	73
6.2.3.	Montaje Electrónico.....	77
6.3	Implementación de interfaz gráfica para el manejo del módulo “SCARA”	82
6.3.1	Diseño Inicial de la Interfaz.....	82
6.3.2	Controles de Movimiento.	83
6.3.3	Funciones de Control.	85
6.3.4	Comunicación con Arduino..	85
6.3.5	Ultimas Mejoras.....	86
6.4	Código de control de la interfaz gráfica	87
6.4.1	Importación de Bibliotecas.	87
6.4.2	Definición de Funciones.	88
6.4.3	Funciones de Movimiento.....	88
6.4.4	Funciones de Control de Secuencias.....	89
6.4.5.	Función de Comunicación con Arduino..	89
6.4.6	Configuración de la Interfaz Gráfica..	90
6.4.7	Finalización y Actualización de la Interfaz.	91
6.5	Código de control del brazo robótico SCARA	91
6.6	Pruebas exhaustivas al prototipo de brazo robótico SCARA	93
6.7	Ajustes Finales.....	98
7.	Conclusiones.....	102
8.	Recomendaciones.....	104
9.	Referencias bibliográficas	107
10.	Bibliografía.....	109
11.	Anexos.....	110

Lista de figuras

	Pág.
Figura 1. Prototipo robot SCARA tipo móvil.....	20
Figura 2. Plataforma robótica de bajo costo	21
Figura 3. Diseño y construcción de un robot manipulador tipo SCARA	22
Figura 4. Diagrama general del sistema.....	23
Figura 5. Robot Puma	28
Figura 6. Robot Esférico	28
Figura 7. RPP	29
Figura 8. Robot Cartesiano	29
Figura 9. Arduino Uno R3	31
Figura 10. Driver para Nema	33
Figura 11. Shield Arduino R3.....	34
Figura 12. Motores NEMA17.....	36
Figura 13. Fuente de alimentación.....	37
Figura 14. Estructura del Robot.....	39
Figura 15. Tornillería.....	40
Figura 16. Correas dentadas.....	42
Figura 17. Metodología.....	45
Figura 18. Varilla lisa	49
Figura 19. Tornillo sin fin Fuente: diseño propio.....	50
Figura 20. Tornillo de ajuste.....	50
Figura 21. Spacers.....	51
Figura 22. Soporte.....	51
Figura 23. Rodamiento pequeño.....	52
Figura 24. Rodamiento lineal.....	52
Figura 25. Rodamiento estándar	53
Figura 26. Polea	53
Figura 27. Pinza	54
Figura 28. Conector	54
Figura 29. Adaptador	55

Figura 30. Conector transmisión.....	55
Figura 31. Conector estructural.....	56
Figura 32. Unión.....	56
Figura 33. Base.....	57
Figura 34. Soporte.....	57
Figura 35. Conector actuador.....	58
Figura 36. Conector soporte.....	58
Figura 37. Nema 17.....	59
Figura 38. Motor servo.....	59
Figura 39. Eje de correa.....	60
Figura 40. Tornillo de ajuste.....	60
Figura 41. Correa pequeña.....	61
Figura 42. Correa grande.....	61
Figura 43. Caja servo.....	62
Figura 44. Brazo extendido izquierdo Fuente: diseño propio.....	62
Figura 45. Brazo extendido izquierdo.....	63
Figura 46. Brazo derecho.....	63
Figura 47. Brazo izquierdo.....	64
Figura 48. Amarre.....	64
Figura 49. Soporte pinza.....	65
Figura 50. Simulación del ensamblaje.....	66
Figura 51. Modelación de correas.....	66
Figura 52. Simulación en Tinkercad.....	68
Figura 53. Parámetros de impresión para correas.....	69
Figura 54. Piezas impresas en PLA sobre la cama de impresión.....	70
Figura 55. Conjunto de piezas adicionales impresas en PLA.....	71
Figura 56. Impresión de correas flexibles en TPU para partes móviles.....	72
Figura 57. Ensamble primer eslabón.....	73
Figura 58. Ensamble segundo eslabón.....	74
Figura 59. Ensamble motor segundo eslabón.....	75
Figura 60. Ensamblaje motor eje Z.....	76

Figura 61. Ensamblaje final	77
Figura 62. Conexiones eléctricas	78
Figura 63. Diagrama eléctrico.....	78
Figura 64. Montaje eléctrico	80
Figura 65. Conexiones eléctricas	81
Figura 66. Montaje eléctrico	82
Figura 67. Plataforma Grafica.....	83
Figura 68. Controles de movimiento	84
Figura 69. Botones enviar y home Fuente: diseño propio	85
Figura 70. Botones guardar, ejecutar y enviar	85
Figura 71. Indicador de estado.....	86
Figura 72. Interfaz Final	87
Figura 73. Importación de Bibliotecas.....	88
Figura 74. Definición de funciones.....	88
Figura 75. Funciones de movimiento.....	89
Figura 76. Control de secuencias	89
Figura 77. Comunicación con Arduino.....	90
Figura 78. Configuración interfaz.....	91
Figura 79. Finalización	91
Figura 80. Finalización	94
Figura 81. Base	95
Figura 82. Pruebas	96
Figura 83. Formato de Preguntas	97
Figura 84. Fotos de las pruebas realizadas con estudiantes	98
Figura 85. Ajustes Finales Brazo Scara	99
Figura 86. Diseño de base y carcasa	100

Lista de tablas

	Pág.
Tabla 1. Parámetros de impresión de piezas estructurales y correas flexibles.....	69
Tabla 2. Controles de movimiento	84

Lista de anexos

	Pág.
Anexo A. Funcionamiento Interfaz Gráfica.....	110
Anexo B. Código Interfaz grafica	111
Anexo C. Código Arduino	112
Anexo D. Pruebas internas	113
Anexo E. Formatos de los estudiantes	114
Anexo F. Fotos y videos de pruebas con estudiantes.....	115
Anexo G. Fotos y videos de Ajustes Finales.....	116

Resumen

DESARROLLO DE UN KIT ROBÓTICO TIPO “SCARA” PARA LA EDUCACIÓN EN REGIONES

ANDERSON ESTEBAN CORREA ARANGO

MARILUZ GARCÍA VERA

ANDERSON LEÓN GIRALDO

Este proyecto se centra en el desarrollo de un kit robótico tipo SCARA accesible y educativo, diseñado para mejorar la enseñanza de la robótica en regiones con recursos limitados. Basado en la experiencia de la Institución Universitaria Pascual Bravo, el proyecto aborda la falta de acceso a tecnología avanzada en zonas rurales, proporcionando una herramienta asequible que permite a los estudiantes interactuar directamente con un sistema robótico. La metodología utilizada combinó diseño CAD, impresión 3D y programación mediante Arduino, lo que resultó en un prototipo funcional y replicable. El kit demostró su capacidad para ejecutar movimientos precisos y su aplicabilidad en contextos educativos, fomentando la formación de habilidades en STEM.

Palabras claves: robótica educativa, SCARA, impresión 3D, aprendizaje STEM, tecnología accesible

Abstract

DEVELOPMENT OF A SCARA-TYPE ROBOTIC KIT FOR EDUCATION IN UNDERRESOURCED REGIONS

ANDERSON ESTEBAN CORREA ARANGO

MARILUZ GARCÍA VERA

ANDERSON LEÓN GIRALDO

This project focuses on the development of an accessible and educational SCARA-type robotic kit, designed to enhance robotics education in regions with limited resources. Based on the experience of the Pascual Bravo University Institution, the project addresses the lack of access to advanced technology in rural areas, providing an affordable tool that allows students to interact directly with a robotic system. The methodology combined CAD design, 3D printing, and Arduino programming, resulting in a functional and replicable prototype. The kit demonstrated its ability to execute precise movements and its applicability in educational contexts, fostering the development of STEM skills.

Keywords: educational robotics, SCARA, 3D printing, STEM learning, accessible technology

Glosario

Actuador: dispositivo mecánico, hidráulico, neumático o eléctrico que produce un movimiento o fuerza en respuesta a una señal de control, como los utilizados para mover las articulaciones del brazo robótico “SCARA”.

Brazo “SCARA”: tipo de robot industrial que se caracteriza por tener dos ejes de rotación y uno lineal, lo que le permite movimientos en un plano horizontal, como un brazo humano, y es comúnmente utilizado en tareas de ensamblaje y pick-and-place en la industria manufacturera.

Interfaz de usuario: sistema o conjunto de herramientas que permite a los usuarios interactuar con el brazo robótico “SCARA”, como una pantalla táctil o un software de control.

Microcontrolador: dispositivo electrónico integrado que contiene una unidad central de procesamiento, memoria y periféricos de entrada/salida, utilizado para el control y la programación del brazo robótico “SCARA”.

Prototipado rápido: proceso de fabricación rápida de piezas o componentes utilizando tecnologías de fabricación aditiva, como la impresión 3D, para construir rápidamente prototipos y modelos del brazo robótico “SCARA”.

Introducción

La educación en robótica se ha convertido en una herramienta esencial para el desarrollo de habilidades tecnológicas y científicas en estudiantes de todo el mundo. Sin embargo, en regiones con recursos limitados, el acceso a laboratorios de tecnología avanzada y equipos especializados es escaso, lo que genera una brecha significativa en la educación tecnológica. La Institución Universitaria Pascual Bravo, comprometida con el desarrollo tecnológico y social de las regiones, ha identificado esta problemática en los municipios donde tiene presencia mediante su programa de Regionalización.

Este proyecto tiene como objetivo desarrollar un kit robótico tipo SCARA (Selective Compliance Articulated Robot Arm) asequible y educativo, diseñado para proporcionar una solución práctica y accesible para la enseñanza de robótica en contextos rurales. El kit SCARA no solo permite a los estudiantes explorar los principios de la robótica y la automatización, sino que también promueve el aprendizaje experiencial y el desarrollo de habilidades en áreas STEM (Ciencia, Tecnología, Ingeniería y Matemáticas).

Mediante una metodología que incluye el diseño asistido por computadora (CAD), la impresión 3D y la programación con Arduino, el proyecto busca ofrecer un prototipo replicable que permita a los estudiantes interactuar directamente con un brazo robótico funcional. Esto facilita el aprendizaje de conceptos fundamentales de la robótica, como la cinemática, el control de motores y el diseño estructural. Además, el proyecto responde a la necesidad de contar con herramientas educativas innovadoras y accesibles en las regiones, promoviendo la equidad en el acceso a la tecnología y la preparación de los estudiantes para un mercado laboral en constante evolución.

1. Planteamiento del problema

1.1 Descripción

La Institución Universitaria Pascual Bravo está comprometida con el desarrollo tecnológico, social y económico no solo de la ciudad de Medellín, sino también de las regiones del departamento de Antioquia donde hace presencia mediante el programa de Regionalización y a través del mismo ofrece actualmente posibilidades de educación en 20 municipios con 7 programas técnicos y tecnológicos que impactan directamente dentro de dichas comunidades.

La problemática que este proyecto aborda se origina en las regiones con recursos limitados, donde la falta de acceso a laboratorios de robótica y tecnología avanzada representa un desafío significativo. En estas regiones, se presenta una brecha educativa que obstaculiza el desarrollo de habilidades tecnológicas esenciales para estudiantes y comunidades. La robótica, una disciplina fundamental en la era actual, no solo impulsa la adquisición de conocimientos en tecnología, sino que también promueve la capacidad de resolver problemas, la innovación y la preparación de individuos para un mercado laboral altamente competitivo.

Este proyecto se basa en la experiencia de la Institución Universitaria Pascual Bravo en la impartición de clases, talleres y semilleros de investigación en regiones con recursos limitados, donde se ha evidenciado la necesidad de recursos educativos y tecnológicos accesibles. La falta de laboratorios de robótica en estas áreas ha limitado la formación en tecnología y ha creado una disparidad en las oportunidades de desarrollo para los estudiantes y las comunidades locales.

La capacitación en robótica no solo impulsa el desarrollo de habilidades, sino que también empodera a las comunidades y estimula la innovación local en contextos con recursos limitados. Por lo tanto, el proyecto tiene relevancia tecnológica, científica y práctica al abordar este problema y proponer una solución en forma de un kit robótico tipo “SCARA” accesible y educativo.

1.2 Formulación

A partir de la problemática antes descrita ¿Cómo diseñar y desarrollar un kit robótico “SCARA” asequible y educativo, adaptado a regiones con recursos limitados, que permita a los estudiantes de las regiones de la Institución Universitaria Pascual Bravo tener un contacto directo con equipos y dispositivos de robótica con el cual se fomente la motivación en la formación investigativa, y cierre la brecha tecnológica en la educación en robótica, superando las limitaciones geográficas y económicas actuales?

2. Justificación

La robótica, como disciplina interdisciplinaria, busca diseñar máquinas robotizadas capaces de realizar diversas tareas automatizadas mediante su software (EDS Robotics , 2021). En el ámbito educativo, la integración de la robótica emerge como un recurso pedagógico esencial para impulsar una educación enfocada en las áreas STEM, generando impactos significativos no solo en el desarrollo educativo, sino también en el ámbito industrial y económico de la sociedad (Ferrada, Carrillo-Rosúa, Díaz-Levicoy, & & Silva-Díaz, 2020).

La actualidad educativa demanda el uso de herramientas pedagógicas innovadoras, no solo por parte de los docentes, sino también de los estudiantes, quienes buscan alternativas que incorporen instrumentos y metodologías novedosas. Este enfoque permite llevar a cabo actividades experimentales que respalden las teorías, facilitando así el desarrollo de ideas, la solución de problemas y la comprensión profunda de los conceptos (Vizcarra Sánchez, Y. A., & Vizcarra Gavilán, A. M., 2021). (Romero Ariza, 2014). El Informe Horizont 2021 para la educación superior, publicado por EDUCAUSE, destaca importantes tendencias relacionadas con la pandemia del COVID-19, como el aprendizaje a distancia, la brecha digital y la salud mental. Desde una perspectiva tecnológica, se observa una marcada inclinación hacia escenarios educativos con modelos híbridos y virtuales mediante el uso de tecnología y telecomunicaciones. Elementos como inteligencia artificial, modelos de cursos mixtos e híbridos, analítica del aprendizaje, microcredenciales, recursos educativos abiertos (REA) y calidad en el aprendizaje en línea son señalados como componentes esenciales. El informe anticipa una tendencia hacia una educación personalizada, flexible y práctica, en consonancia con la disminución de recursos económicos disponibles para la inversión en educación (EDUCASE, 2021).

El Ministerio de Educación Nacional de Colombia, a través de su oficina de innovación educativa con uso de TIC, promueve programas para implementar herramientas tecnológicas al servicio de la educación a distancia. Estos programas buscan el desarrollo de comunidades del conocimiento adaptadas al contexto de aprendizaje, posibilidades, retos y recursos disponibles en el país (Mineducación, 2004).

La propuesta de desarrollar entrenadores portátiles aplicados a la robótica para los estudiantes en las regiones se presenta como una respuesta a las limitaciones existentes en el acceso directo a elementos físicos y dispositivos reales. Este enfoque tiene como objetivo cerrar la brecha tecnológica y mejorar la calidad de la formación investigativa, al proporcionar una herramienta que facilite la interacción directa con los elementos de trabajo y promueva el aprendizaje experiencial (Rodríguez, 2009) . La experimentación física, como método de aprendizaje significativo, se convierte en un instrumento fundamental para que los estudiantes comprendan teorías y las apliquen en la solución de problemas específicos.

El objetivo principal de este estudio consiste en desarrollar un prototipo de kit robótico tipo “SCARA” asequible, educativo y fácil de usar, destinado a la enseñanza y formación de las regiones en el campo de la robótica y la automatización. Todo ello a partir de un proceso organizado de diseño, desarrollo y evaluación que permita obtener un prototipo que pueda ser replicado mediante los recursos del proyecto principal que cubre este trabajo de grado.

En consecuencia, la propuesta de desarrollar entrenadores portátiles aplicados a la robótica para los estudiantes en las regiones se alinea con las tendencias educativas actuales y busca mejorar la calidad educativa, cerrar la brecha tecnológica y proporcionar soluciones competentes para abordar las problemáticas específicas del contexto regional. Este enfoque contribuirá al desarrollo de competencias, facilitará el aprendizaje y la calidad de la educación, promoverá el contacto con dispositivos industriales y fortalecerá la estrategia de investigación formativa en las regiones donde la institución tiene presencia (Mineducación, 2004).

Además, se espera que este enfoque no solo brinde beneficios educativos, sino que también tenga un impacto positivo a nivel social, tecnológico, teórico y ambiental. Al ser una herramienta inclusiva, promoverá la equidad en el acceso a la educación en robótica, preparará a los estudiantes para los desafíos tecnológicos actuales y futuros, enriquecerá la teoría con experiencias prácticas y fomentará una mentalidad sostenible al abordar problemas ambientales mediante soluciones tecnológicas innovadoras y sostenibles.

3. Objetivos

3.1 Objetivo general

Desarrollar un kit robótico tipo "SCARA" con interfaz gráfica asequible, educativo y fácil de usar, mediante técnicas de prototipado rápido de productos, implementación electrónica y robotización, destinado a la enseñanza y formación en los campos de la robótica y la automatización, de investigadores y estudiantes de las regiones donde la Institución Universitaria Pascual Bravo hace presencia.

3.2 Objetivos específicos

Diseñar un prototipo de brazo robótico "SCARA" mediante herramientas CAD, optimizando tanto la eficiencia, como los costos que garanticen su accesibilidad.

Desarrollar un prototipo de brazo robótico "SCARA" replicable, utilizando materiales asequibles y técnicas de fabricación eficientes, para su implementación en contextos educativos.

Implementar una interfaz gráfica para el manejo del módulo "SCARA" mediante Python, que permita una operación intuitiva y eficiente del brazo robótico en entornos educativos.

Realizar pruebas exhaustivas al prototipo de brazo robótico SCARA, por medio de la evaluación de su precisión, velocidad y estabilidad, que permita la identificación de posibles mejoras y optimizaciones.

4. Marco teórico

4.1 Estado del arte

Antes de establecer los referentes teóricos en los que se fundamenta el proyecto, se presentarán algunos trabajos similares a este proyecto, indicando qué hicieron, cómo lo hicieron, qué resultados obtuvieron y cómo se relaciona con el desarrollo del kit robótico tipo “SCARA” para la educación en regiones.

Aguirre, Orozco, Lara y Luna (2022) desarrollaron el proyecto "Prototipo de Robot SCARA con Base Móvil," centrado en la aplicación de robótica para la enseñanza. Utilizaron herramientas de diseño CAD para modelar un brazo robótico SCARA con una base móvil, y emplearon tanto cinemática directa como inversa para controlar su movimiento. Además, implementaron una interfaz gráfica sencilla, facilitando la interacción del usuario con el robot. Los resultados demostraron que es posible crear un sistema educativo y práctico, adaptable a distintos niveles de formación, lo que es crucial para la educación en robótica.

Este trabajo tiene una relación significativa con el proyecto tratado en este documento que consisten en el desarrollo de un kit robótico SCARA asequible y educativo. Ambos comparten el enfoque en herramientas accesibles para fomentar el aprendizaje, subrayando la posibilidad de integrar tecnologías asequibles como CAD y controladores simples para fines educativos. Este proyecto refuerza la viabilidad de los sistemas SCARA en la enseñanza y muestra que estos prototipos pueden contribuir a la mejora de la comprensión de la robótica en diversos contextos educativos (Aguirre Cerrillo, Orozco Magdaleno, Lara González, & Luna Castro, 2016).

En la siguiente figura se presenta una imagen de los resultados obtenidos en este proyecto, es de notar que se trata de un desarrollo académico de un prototipo funcional, desarrollado con insumos comerciales y algunos materiales comunes como madera, con la respectiva adaptación del sistema neumático.

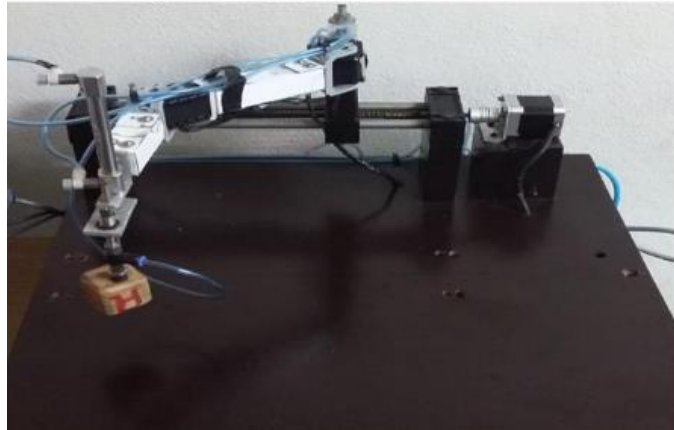


Figura 1. Prototipo robot SCARA tipo móvil

Fuente: extraído de <https://mdi.uanl.mx/index.php/revista/article/view/122/102>

Soriano, Marín, Juan, Cazalilla, Valera, Vallés y Albertos (2013) presentan en su trabajo titulado "Plataforma Robótica de Bajo Coste y Recursos Limitados Basada en Arduino y Dispositivos Móviles" un enfoque innovador para la educación en robótica. Este proyecto utiliza componentes de bajo costo, haciendo accesible la enseñanza de la robótica en contextos con recursos limitados, especialmente en lugares que carecen de laboratorios costosos. La plataforma permite a los estudiantes experimentar con programación y electrónica, fomentando habilidades prácticas y teóricas en robótica, lo cual es fundamental para la formación técnica en diversas áreas. Su trabajo destaca la importancia de la accesibilidad en la educación tecnológica, ya que ofrece una solución viable para la enseñanza en entornos desfavorecidos.

Esta investigación es relevante para este proyecto porque proporciona un modelo de implementación de tecnologías accesibles en la educación, mostrando que es posible desarrollar soluciones efectivas y económicas para el aprendizaje de la robótica en cualquier entorno (Soriano , y otros, 2013).

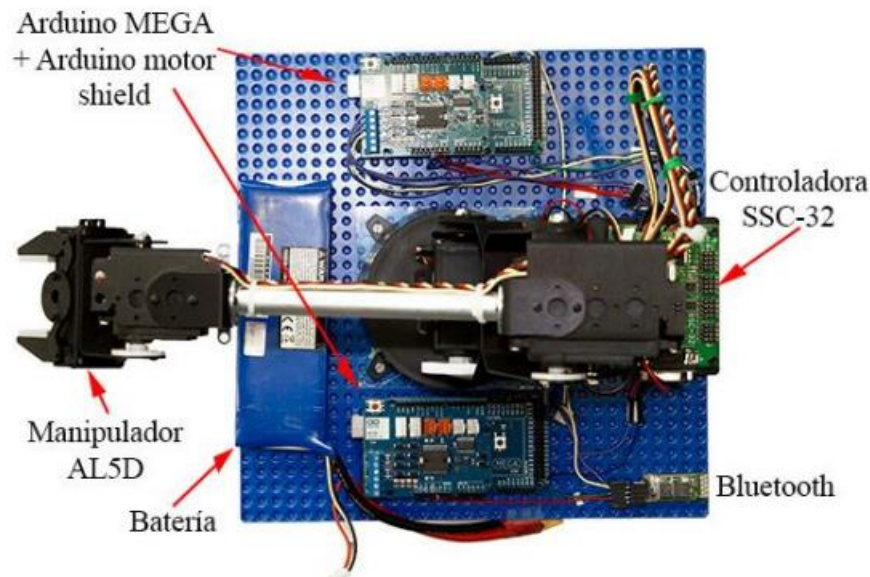


Figura 2. Plataforma robótica de bajo costo

Fuente: extraído de http://wks.gii.upv.es/cobami/files/MV_JJAAPlataforma%20Arduino.pdf

Jiménez (2009), en su trabajo titulado "Diseño y construcción de un robot manipulador tipo SCARA con sistema de visión de máquina, que pueda ser utilizado como herramienta didáctica en asignaturas de robótica y control", presenta una plataforma de aprendizaje que aborda conceptos teóricos y prácticos de robótica y control. El proyecto consiste en un manipulador SCARA de 4 grados de libertad (DOF), diseñado para ser un recurso didáctico en la enseñanza de estas materias.

La investigación incluye una descripción detallada del diseño mecánico, electrónico e informático del robot, así como la validación de un modelo matemático mediante varias técnicas de control. El sistema está equipado con un módulo de visión que permite al robot identificar y manipular objetos dentro de su espacio de trabajo, facilitando tareas automatizadas sin supervisión directa. Se desarrolló un software educativo que permite a los usuarios realizar análisis de variables articulares, configurar trayectorias de movimiento y calibrar métodos de control. Además, se elaboraron video-manuales que explican la simulación, configuración y puesta en marcha del robot, haciendo la plataforma accesible incluso para aquellos sin una formación técnica avanzada.

Este trabajo es crucial para el proyecto que se está desarrollando porque demuestra cómo se pueden integrar conceptos teóricos de robótica en un sistema práctico, facilitando el aprendizaje en contextos donde no hay acceso a laboratorios costosos. Además, las metodologías y recursos desarrollados pueden ser adaptados para el diseño y construcción de nuestro kit robótico tipo SCARA, asegurando su relevancia y aplicabilidad en la educación técnica en regiones con recursos limitados (Jimenez Robayo , 2009).



Figura 3. Diseño y construcción de un robot manipulador tipo SCARA

Fuente: extraído de <https://repository.unimilitar.edu.co/items/c647ab58-ba68-4e04-98ea-a1f705e756ca>

Estos estudios ofrecen un panorama claro sobre las metodologías y tecnologías empleadas en el desarrollo de brazos robóticos educativos y portátiles. Al analizar estos trabajos, este proyecto "Desarrollo de un kit robótico tipo SCARA asequible, educativo y fácil de usar para la enseñanza y formación en regiones" puede beneficiarse de sus hallazgos y enfoques, asegurando que el kit robótico tipo "SCARA" no solo sea asequible y educativo, sino también altamente funcional y accesible para las regiones objetivo.

4.2 Referentes teóricos

La robótica es un área de aplicación de diferentes técnicas de diseño y construcción de aparatos o robots para efectuar trabajos u operaciones, en la mayoría de los casos en zonas industriales para la sustitución de la mano de obra humana. A continuación, en el siguiente

diagrama se presenta un recuento de los principales fundamentos teóricos que sostendrán el desarrollo de este proyecto desde los objetivos planteados.

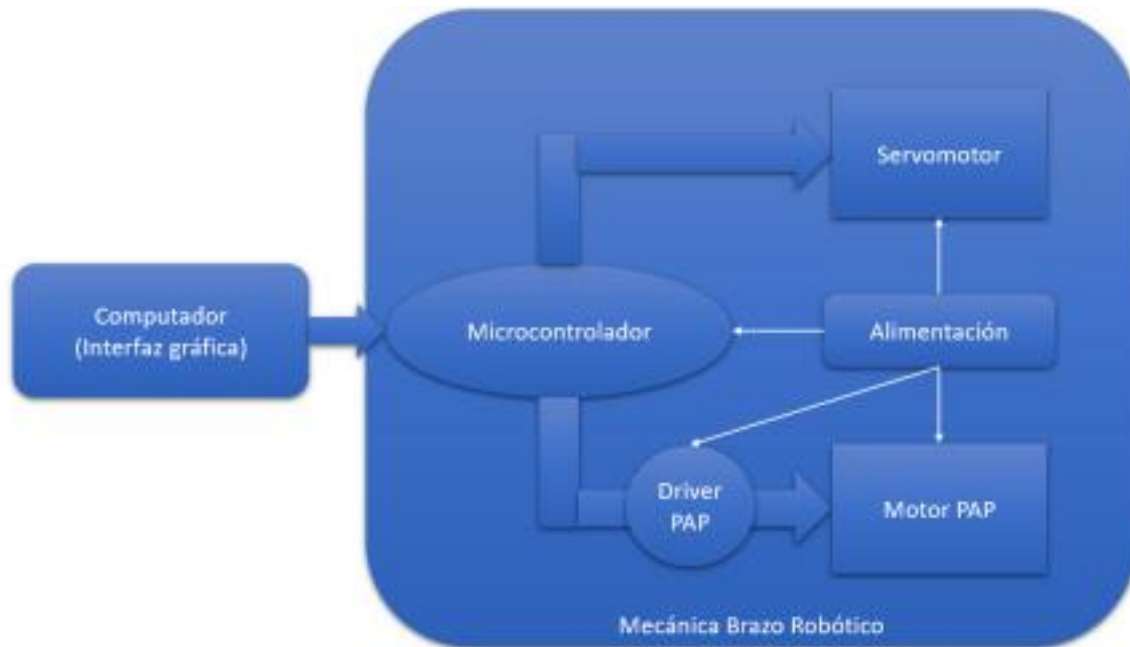


Figura 4. Diagrama general del sistema
Fuente: diseño propio

4.2.1 Términos importantes. A continuación, se exponen algunos términos y conceptos teóricos con el fin de hacer claridad en el tema y tener un panorama general del mismo.

A. Manipulador: es un sistema de cuerpos (eslabones) conectado por medio de articulaciones. Este sistema forma una cadena cinemática que contiene un eslabón fijo llamado base y un eslabón de salida, en el cual es fijada una herramienta, llamada efector. Un manipulador puede ser de cadena abierta o cerrada (también conocido como manipulador paralelo) y opera en un espacio plano o tridimensional de acuerdo con la tarea programada (Cortés, 2020).

B. Cadena cinemática: es la estructura del robot la cual está constituida por eslabones o uniones de eslabones y se clasifica en dos tipos: cadena abierta y cadena cerrada (Cortés, 2020).

C. Cadena abierta: cuando el dispositivo robótico cuenta con un conjunto determinado de articulaciones y eslabones y se llega a un punto en el cual el último eslabón no se encuentra conectado a ningún otro elemento. Un ejemplo de cadena abierta es el robot UNIVERSAL y el robot PUMA (Cortés, 2020).

D. Cadena cerrada: cuando el dispositivo robótico cuenta con un conjunto determinado de articulaciones y eslabones que están conectados a dos o más eslabones. Un ejemplo de cadena cerrada es el robot DELTA (Cortés, 2020).

E. Manipulador serial: son los manipuladores de cadena abierta; es decir, sus elementos son conectados uno a uno en serie (Cortés, 2020).

F. Manipulador paralelo: son los manipuladores de cadena cerrada, en la cual se pueden identificar varios lazos, como elementos en paralelo (Cortés, 2020).

G. Grados de libertad: son los parámetros que se precisan para determinar la posición y la orientación del efecto final del manipulador. También se pueden definir los grados de libertad como los posibles movimientos básicos (giratorios y de desplazamiento) independientes. Un mayor número de grados de libertad conlleva un aumento de la flexibilidad en el posicionamiento del efector final. Aunque la mayoría de las aplicaciones industriales requieren 6 grados de libertad como las de la soldadura, mecanizado y palatización, otras más complejas reciben un número mayor (Cortés, 2020).

H. Carga útil: es el peso que un robot puede cargar, o sea la capacidad de carga máxima que el robot puede manipular sin presentar daños en su estructura. Sobrepasando la carga máxima el robot puede volverse menos preciso, puede no conseguir su trayectoria con precisión, o puede tener desvíos excesivos. La carga útil comparada con el peso del robot, es generalmente muy pequeña. Por ejemplo, el robot Fanuc Robotics LR Mate tiene un peso mecánico de 86 lb y una carga útil de 6,6 lb y el robot M-16 tiene un peso mecánico de 594 lb y una carga útil de 35 lb (Cortés, 2020).

I. Alcance: es la distancia máxima que un robot puede alcanzar dentro de su espacio de trabajo. Muchos puntos dentro del espacio de trabajo del robot pueden ser alcanzados con cualquier orientación deseada. En otros puntos sobrepasa el límite de la capacidad de alcance y la orientación no puede ser especificada. El alcance es una función de las articulaciones y movimientos del robot y de su respectiva configuración. Esta es una especificación importante para robots industriales y debe ser considerada antes de ser seleccionado o instalado el robot (Cortés, 2020).

J. Precisión: es definida como la exactitud con que un determinado punto puede ser alcanzado. Esta es una función de la resolución de los actuadores. La mayoría de los robots industriales puede tener la precisión en un rango de 0,001 pulgadas o menos. La precisión es una función de cuantas posiciones y orientaciones fueron usadas para testear el robot, con que carga, y en que velocidad. Cuando la precisión es una especificación importante, es crucial investigar esas cuestiones (Cortés, 2020).

K. Repetibilidad: es la precisión con que la misma posición puede ser alcanzada si el movimiento es repetido muchas veces. Suponga que un robot es accionado para el mismo punto 100 veces. Una vez que muchos factores pueden afectar la precisión de la posición, el robot no puede llegar al mismo punto cada vez, más quedará dentro de un determinado radio a partir del punto deseado. El radio de un círculo formado por los movimientos repetidos es llamado repetibilidad. La repetibilidad es mucho más importante que la precisión. Si un robot no es preciso, en general muestra un error consistente, que puede ser previsto y, por lo tanto, corregido por programación. Por ejemplo, suponga que un robot está siempre fuera de 0,05 pulgadas para la derecha. En ese caso, todos los puntos deseados pueden ser especificados en 0,05 pulgadas a la izquierda y, así, eliminar el error. No en tanto, sí el error es aleatorio, no puede ser previsto y, consecuentemente, no puede ser eliminado. La repetibilidad define la extensión de ese error aleatorio, es generalmente especificada para un determinado número de corridas. Un buen test proporciona mayores resultados con datos más reales. Los fabricantes deben especificar la repetibilidad en conjunto con el número de test, la carga útil aplicada durante el test, y la orientación del brazo. Por ejemplo, la repetibilidad de un brazo en una dirección vertical será diferente cuando el brazo es testado en una configuración horizontal. La

mayoría de los robots industriales tiene repetibilidad en el rango de 0,001 pulgadas (Castillo Castañeda & Rodríguez Vázquez, 2011).

L. Espacio de trabajo de un robot: dependiendo de su configuración y del tamaño de sus eslabones y articulaciones, los robots pueden llegar a una región de puntos alrededor de ellos, que constituyen el espacio de trabajo. La forma de espacio de trabajo para cada robot está exclusivamente relacionada con la estructura. El espacio de trabajo puede ser encontrado matemáticamente escribiendo ecuaciones que definen los eslabones y articulaciones del robot y esos incluye sus limitaciones, como los alcances de los movimientos de cada articulación. Como alternativa, el espacio de trabajo puede ser encontrado empíricamente por mover virtualmente cada articulación por medio de su gama de movimiento, combinando todo el espacio que puede alcanzar, y sustrayendo el que no puede alcanzar. Cuando un robot es considerado para una determinada aplicación, su espacio de trabajo debe ser estudiado para garantizar que el robot sea capaz de alcanzar los puntos deseados. Para la determinación exacta del espacio de trabajo, consultar las hojas de datos de los fabricantes. Lenguaje de máquina: en ese nivel, los programas son escritos en lenguaje de máquina. Este nivel de programación es el más básico y es muy eficiente, pero es más difícil de comprender y entender. Todos los lenguajes acabaron por ser interpretados o compilados por este nivel. No obstante, en el caso de programas de nivel superior, el usuario escribe el programa en lenguajes de alto nivel que son más fáciles de comprender y entender. Nivel punto a punto: En ese nivel, las coordenadas de los puntos son insertadas secuencialmente, y el robot sigue los puntos, conforme lo especificado. Este es un tipo muy primitivo y simple de programación, que es fácil de usar, más no muy poderoso. Carece igualmente de desvíos, de información sensorial y de instrucciones condicionales (Jara, 2008).

M. Nivel primitivo de movimiento: en esos lenguajes, es posible desarrollar un programa más sofisticado, incluyendo la información sensorial, desvíos e instrucciones condicionales. La mayoría de lenguaje de este nivel son interpretadas (Cortés, 2020).

N. Nivel de programación estructurada: la mayoría de los lenguajes en este nivel son compiladas, son poderosas, y permiten una programación más sofisticada. No en tanto, ellas también son más difíciles de aprender (Cortés, 2020).

O. Nivel orientado a la tarea: no existen lenguajes reales en este nivel todavía, la propuesta de IBM en la década de 1980 nunca se materializó, consistía en que en vez de programar un robot para ejecutar una tarea programando cada paso necesario para completarla, el usuario debería solo apenas mencionar la tarea, en cuanto el controlador criara la secuencia necesaria. Imagine que un robot tenga que clasificar tres cajas por tamaño. En todos los lenguajes existentes, el programador tendrá que decir al robot exactamente qué hacer, por lo tanto, cada paso debe ser programado. El robot debe ser informado como ir para la caja mayor, como coger la caja, donde colocarla, donde coger la caja siguiente, y así por lo consiguiente. Con la propuesta de IBM, el usuario apenas indicaría “clasificar”, en cuanto el controlador del robot crearía automáticamente la secuencia. Eso nunca aconteció (Cortés, 2020).

4.2.2 Tipos de articulaciones. Existen diferentes tipos de articulación de brazos robóticos, sin embargo, dos de ellas son más pertinentes analizar para este proyecto.

Articulación rotacional: articulación con un grado de movilidad que permite un movimiento relativo de rotación alrededor de un eje de rotación común, normalmente consistente en una junta giratoria manejada por motores eléctricos y transmisiones, o por cilindros hidráulicos y palancas (Caparroso, 1999).

Articulación prismática: articulación con un grado de movilidad que permite movimiento relativo de traslación rectilínea a lo largo del eje de la articulación (Caparroso, 1999).

4.2.3 Clasificación de manipuladores robóticos. Existen muchas posibilidades de clasificar a los robots industriales. Por ejemplo, tomando en cuenta la aplicación del robot, a través del tipo de control empleado, por la cadena cinemática o geometría del robot. Considerando la geometría, las primeras tres articulaciones que le brindan el posicionamiento

al robot, permiten clasificar al robot como manipuladores rotacionales RRR, esféricos RRP, cilíndricos RPP y manipuladores cartesianos PPP (Caparroso, 1999).

Manipuladores RRR: en algunas ocasiones se le denominan manipuladores rotacionales o antropomórficos. Donde las tres primeras articulaciones son rotacionales. Un ejemplo de dichos manipuladores es el Robot Puma (Figura 2). Una de sus ventajas es que permite una amplia variedad de movimientos en espacios confinados (Caparroso, 1999).



Figura 5. Robot Puma

Fuente: extraído de <https://www.logicbus.com.mx/blog/puma-el-pionero-en-robots-de-ensamblaje/>

Manipuladores RRP: se denominan manipuladores esféricos. Su nombre se debe a que la posición del efector final se puede describir por medio de coordenadas esféricas. Un ejemplo de este tipo de robots es el manipulador Stanford (Figura 2). En estos manipuladores, las dos primeras articulaciones son rotacionales, mientras que la tercera es prismática (Caparroso, 1999).



Figura 6. Robot Esférico

Fuente: extraído de <https://www.logicbus.com.mx/blog/puma-el-pionero-en-robots-de-ensamblaje/>

Manipuladores RPP: estos manipuladores se definen como cilíndricos. Para estos robots, las variables de las articulaciones pueden expresarse a través de coordenadas cilíndricas del efector final referido a la base. En la Figura 3, se observa que la primera articulación es rotacional, mientras que las siguientes son prismáticas (Caparroso, 1999).

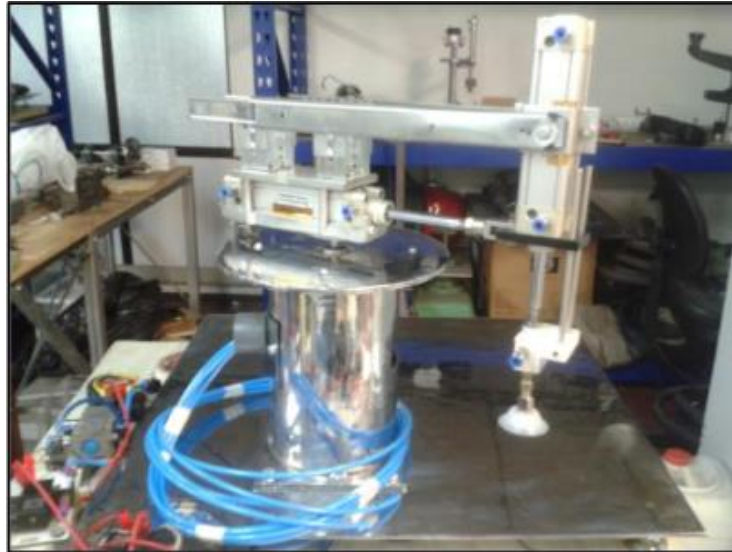


Figura 7. RPP

Fuente: extraído de <https://app.eam.edu.co/ojs/index.php/ingeam/article/download/52/79>

Manipuladores PPP: también definido como manipuladores cartesianos, las variables de las articulaciones pueden manejarse mediante coordenadas cartesianas para referir el efector final con respecto a su base. Las primeras tres articulaciones de los mismos son prismáticas. Entre sus aplicaciones se encuentran el ensamblaje y transferencia de material (Caparroso, 1999).



Figura 8. Robot Cartesiano

Fuente: extraído de <https://marcomponentes.es/tienda/cartesianos/>

4.2.4 Componentes eléctricos y electrónicos. Ahora, se presentan las características de los diferentes dispositivos necesarios para la construcción del prototipo, relacionados con el control del mismo.

A. Placa Arduino Uno y Cable Micro-USB: la placa Arduino Uno junto con su cable micro-USB actúan como el cerebro del robot, controlando todas las operaciones y funciones del mismo (Guedes, 2024).

- Placa compatible con Arduino Uno.
- **Cable micro-USB:** utilizado para la alimentación de la placa y la comunicación con la computadora.
- **Microcontrolador:** equipado con el ATmega328P, que se encarga de ejecutar las instrucciones programadas.
- **Velocidad de procesamiento:** opera a una frecuencia de reloj de 16 MHz.
- **Voltaje de operación:** funciona con un voltaje de 5V.
- **Rango de voltaje de entrada:** acepta una entrada de entre 7.5 y 12 voltios.
- **Conectores de entrada/salida:** dispone de 14 pines digitales, de los cuales 6 pueden ser utilizados para salidas PWM, y 6 pines analógicos para lectura de sensores.
- **Comunicación serie:** incluye un puerto serie por hardware para facilitar la comunicación con otros dispositivos.
- **Memoria interna:** cuenta con 32 KB de memoria Flash (0,5 KB reservados para el bootloader), 2 KB de RAM y 1 KB de EEPROM.

Características adicionales

- **Compatibilidad:** soporta bibliotecas y shields de Arduino, lo que permite la expansión de sus capacidades.
- **Dimensiones:** 68.6 mm x 53.4 mm, lo que la hace adecuada para proyectos de tamaño medio.
- **Corriente máxima de los pines:** los pines de entrada/salida pueden proporcionar hasta 40 mA, permitiendo la conexión directa de LEDs y otros dispositivos de baja corriente.

- **Sistema de regulación de voltaje:** incluye un regulador de voltaje integrado que protege el microcontrolador y otros componentes de picos de corriente.
- **Facilidad de programación:** compatible con el entorno de desarrollo de Arduino (IDE), lo que facilita la programación y depuración de código.
- **Indicadores LED:** dispone de LEDs indicadores para facilitar la visualización del estado de la placa, como el encendido y la transmisión de datos.

La placa Arduino Uno proporciona la capacidad de programar y controlar el brazo robótico, mientras que el cable micro-USB permite la conexión con la computadora para la programación y la transferencia de datos.

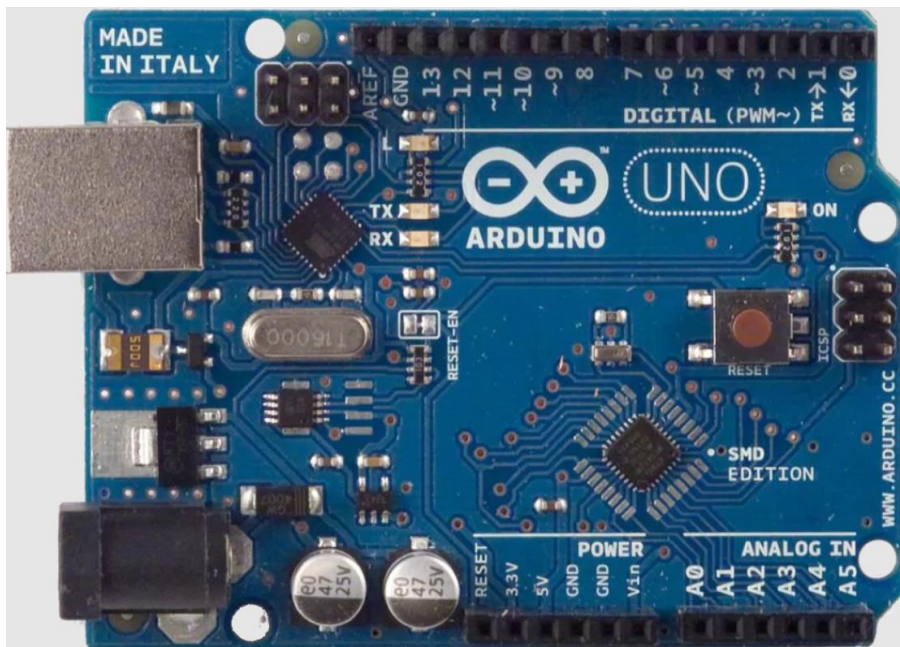


Figura 9. Arduino Uno R3

Fuente: extraído de <https://www3.gobiernodecanarias.org/medusa/ecoblog/rsuagued/arduino/>

B. Driver A4988: es un controlador de motores paso a paso que permite el control preciso del movimiento de los motores del brazo robótico (Jaramillo, 2024).

- **Control de motores paso a paso:** el A4988 está diseñado específicamente para controlar motores paso a paso, permitiendo ajustes finos en el movimiento.

- **Diseño compacto:** su pequeño tamaño facilita su integración en sistemas electrónicos, ocupando poco espacio.
- **Tensión de operación:** funciona con una tensión de 3 a 5.5 VDC para la lógica del controlador y de 8 a 35 VDC para alimentar el motor paso a paso.
- **Capacidad de corriente:** soporta una corriente máxima de 2 A por bobina del motor, permitiendo el control de motores de tamaño moderado.
- **Resolución de pasos:** ofrece múltiples configuraciones de microstepping, incluyendo paso completo, 1/2 paso, 1/4 paso, 1/8 paso, y 1/16 paso, lo que permite ajustar la precisión y suavidad del movimiento del motor.
- **Pines de control:** dispone de pines específicos para controlar el paso (STEP) y la dirección (DIRECTION) del motor, facilitando la programación del movimiento.

Características adicionales

- **Protecciones integradas:** incluye protección contra sobrecorriente, sobretensión y corto circuito, lo que aumenta la durabilidad del controlador y del motor.
- **Ajuste de corriente:** dispone de un potenciómetro para regular la corriente suministrada al motor, permitiendo ajustar el rendimiento y evitar el sobrecalentamiento.
- **Modo de reposo:** tiene una función de reposo (sleep mode) que reduce el consumo de energía cuando el motor no está en uso.
- **Facilidad de uso:** compatible con la mayoría de los motores paso a paso bipolares, lo que lo hace versátil y fácil de implementar en distintos proyectos.
- **Conexión estándar:** utiliza un pinout común, facilitando su integración con placas de desarrollo como Arduino.

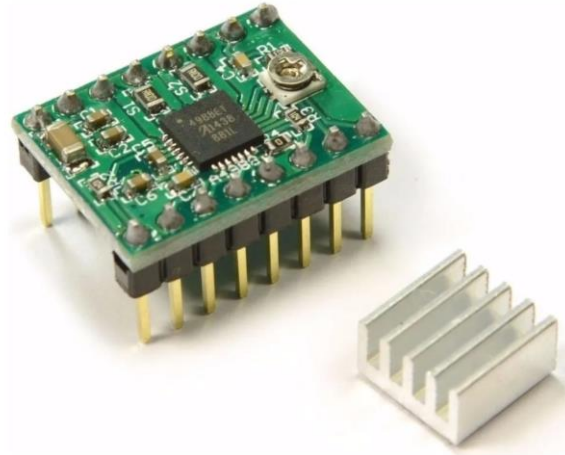


Figura 10. Driver para Nema

Fuente: extraído de <https://www.diarioelectronicohoy.com/blog/descripcion-del-driver-a4988>

Los drivers A4988 se utilizan para controlar los motores paso a paso que accionan las articulaciones del brazo robótico, permitiendo un movimiento preciso y controlado.

C. Shield CNC para Arduino Uno: esta placa de expansión se conecta a la placa Arduino Uno, ampliando sus capacidades y permitiendo el control de hasta 4 motores paso a paso, lo que la hace ideal para aplicaciones como el movimiento de las articulaciones de un brazo robótico.

- **Modelo:** shield CNC 3.0.
- **Ampliación de capacidades:** expande las funciones de la placa Arduino Uno, permitiendo el control simultáneo de hasta 4 motores paso a paso.
- **Control de ejes:** puede manejar 4 ejes diferentes (X, Y, Z, A), lo que ofrece gran flexibilidad en proyectos de automatización y robótica.
- **Finales de carrera:** incluye soporte para 6 finales de carrera, 2 por cada uno de los ejes principales (X, Y, Z), lo que permite establecer límites de movimiento precisos para el brazo robótico.
- **Alimentación:** requiere una fuente de alimentación de 12 a 36V DC, dependiendo de los controladores de motor que se utilicen.
- **Compatibilidad con software:** es compatible con el software GRBL, que es ampliamente utilizado para el control de CNC y facilita la programación del movimiento de los motores.

- **Configuración de microstepping:** dispone de jumpers que permiten ajustar el microstepping, brindando un control más preciso del movimiento de los motores.
- **Conectores de motor:** los motores se conectan mediante bornes tipo Molex de 4 pines, lo que asegura una conexión sólida y confiable.

Características adicionales

- **Dimensiones:** la placa tiene unas dimensiones de 69 mm x 53 mm x 19 mm, lo que la hace compacta y fácil de integrar en proyectos que utilizan la Arduino Uno.
- **Peso:** ligera, con un peso de solo 33 gramos, lo que minimiza el impacto en el peso total del sistema.
- **Salidas adicionales:** incluye salidas para control de dirección del husillo ("Spindle enable") y para el sistema de refrigeración ("Coolant enable"), ampliando aún más las posibilidades de control en aplicaciones CNC.
- **Facilidad de uso:** diseñada para ser compatible con la Arduino Uno, lo que simplifica la instalación y configuración en proyectos de robótica y CNC.

El Shield CNC es fundamental para controlar los motores paso a paso que mueven las articulaciones del brazo robótico, proporcionando una interfaz conveniente y poderosa entre la placa Arduino Uno y los motores, permitiendo un control preciso y eficiente.

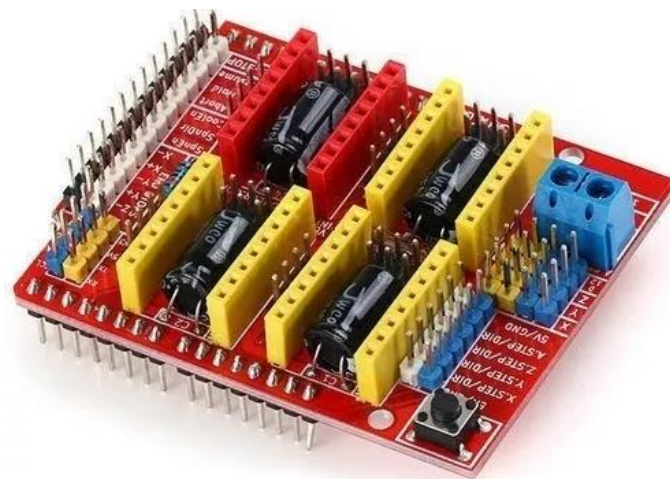


Figura 11. Shield Arduino R3

Fuente: extraído de <https://www.innovatronics.com.co/MCO-484835114-shield-cnc-para-arduino-uno-v3- JM>

D. Motores NEMA17: estos son motores paso a paso de alto rendimiento, utilizados para proporcionar movimiento preciso y controlado en las articulaciones del brazo robótico.

- **Tipo de motor:** NEMA17, con un tamaño de 40 mm.
- **Fases:** motor de 2 fases, lo que permite un control preciso del movimiento.
- **Modelo:** 17HD40005H-22B.
- **Ángulo de paso:** cada paso del motor corresponde a un ángulo de 1.8 grados, lo que proporciona alta precisión en el posicionamiento.
- **Voltaje nominal:** opera a un voltaje nominal de 2.2 V, adecuado para aplicaciones de bajo voltaje.
- **Potencia nominal:** la potencia nominal del motor es de 3.52 W, lo que le permite manejar cargas moderadas.
- **Corriente nominal:** requiere una corriente nominal de 1.66 A, necesaria para alcanzar su torque nominal.
- **Torque nominal:** proporciona un torque de 3.22 N-m, lo que permite mover cargas considerables y mantener la estabilidad en las articulaciones del brazo robótico.
- **Resistencia:** tiene una resistencia de 1.65 Ohms, lo que influye en la corriente que circula a través del motor.
- **Inductancia:** la inductancia es de 2.5 mH, un valor que afecta la respuesta dinámica del motor.
- **Diámetro del eje:** el eje del motor tiene un diámetro de 5 mm, compatible con una amplia gama de acoplamientos y engranajes.
- **Dimensiones:** sus dimensiones son 42 x 42 x 40 mm, lo que lo hace compacto y fácil de integrar en diferentes proyectos de robótica.

Los motores NEMA17 se emplean en el proyecto para generar el movimiento necesario en las articulaciones del brazo robótico, proporcionando un control preciso sobre la posición y la velocidad. Esto es esencial para lograr movimientos suaves y controlados, fundamentales en aplicaciones de robótica donde la precisión es clave. Además, su alto torque y diseño compacto los hacen ideales para aplicaciones que requieren fiabilidad y durabilidad.



Figura 12. Motores NEMA17

Fuente: extraído de <https://www.applied-motion.com/s/article/what-do-nema-sizes-mean>

E. Fuente de Alimentación de 12V a 4A: esta fuente de alimentación está diseñada para proporcionar la energía necesaria para operar todos los componentes del brazo robótico, incluyendo los motores y la placa Arduino Uno, garantizando un funcionamiento estable y confiable.

- **Salida:** proporciona una salida de 12V y 4A, suficiente para alimentar múltiples componentes simultáneamente.
- **Diseño:** compacto y eficiente, facilitando su integración en diferentes entornos de trabajo.

Características adicionales

- **Tecnología:** utiliza tecnología de switching con salida regulada y función de cargador de flote, lo que permite una regulación eficiente de la tensión de salida y una carga continua y estable de las baterías conectadas.
- **Tensión de salida:** la tensión de salida es de 13.8V DC, con capacidad de ajuste en un rango de $\pm 10\%$, adaptándose a diferentes necesidades de alimentación.
- **Corriente de carga DC:** ofrece una corriente de carga de 0.35A para dispositivos conectados, y una salida de corriente DC de 4.0A, proporcionando suficiente energía para componentes que requieren altos niveles de corriente.

- **Rango de entrada:** admite un amplio rango de entrada de 90 a 245V AC, funcionando en frecuencias de 47 a 63 Hz, lo que permite su uso en diversas regiones con distintas configuraciones eléctricas.
- **Eficiencia:** cuenta con una eficiencia del 81%, minimizando pérdidas de energía y mejorando el rendimiento general.
- **Ripple máximo:** mantiene un nivel máximo de ripple de 120mVp-p, asegurando una salida de energía limpia y estable.
- **Protecciones:** incluye protección contra cortocircuitos y sobrecorriente, asegurando la seguridad de los componentes conectados y la durabilidad del equipo.
- **Tiempo de conmutación:** tiene un tiempo de conmutación de 200 ms, garantizando una respuesta rápida a cambios en la demanda de energía.
- **Protección de baja batería:** implementa protección de baja batería, evitando la descarga completa y extendiendo la vida útil de las baterías.
- **Temperatura de trabajo:** opera en un rango de temperatura de -10°C a 60°C, lo que la hace apta para su uso en diferentes entornos.
- **Dimensiones:** sus dimensiones son 129 x 98 x 38 mm, lo que la convierte en una opción compacta y fácil de instalar en proyectos donde el espacio es limitado.

Esta fuente de alimentación es clave para el funcionamiento del brazo robótico, asegurando que todos los componentes reciban la energía necesaria de manera constante y segura.



Figura 13. Fuente de alimentación

Fuente: extraído de <https://suconel.com/product/fuente-de-alimentacion-suichada-de-4-salidas-dc-5v-12v-24v-12v/>

4.2.5 Componentes mecánicos. Finalmente, se presentan las características de los diferentes dispositivos necesarios para la construcción del prototipo, relacionados con los elementos mecánicos que garantizan el movimiento del mismo.

A. Estructura del robot: la estructura del robot es esencial para proporcionar el soporte físico y la rigidez necesarios para mantener todas las partes en su lugar y permitir el movimiento coordinado del brazo robótico.

- **Material:** fabricada mediante impresión 3D utilizando plástico PLA, que ofrece una buena relación entre rigidez y flexibilidad, además de ser ligero y fácil de imprimir.
- **Dimensiones:** el tamaño de la estructura puede variar según el diseño específico, pero generalmente está optimizado para albergar todos los componentes mecánicos y electrónicos de manera compacta y eficiente.
- **Peso:** el peso total de la estructura impresa en PLA suele ser ligero, lo que minimiza la carga sobre los motores del brazo robótico.
- **Capacidad de carga:** diseñada para soportar los componentes electrónicos, motores y otros elementos, garantizando que la estructura no se deforme bajo el peso y las fuerzas operativas.
- **Montaje:** la estructura incluye puntos de anclaje y perforaciones precisas para facilitar el montaje de motores, ejes, poleas y demás componentes mecánicos y electrónicos.
- **Resistencia térmica:** el material PLA tiene una temperatura de deformación alrededor de los 60-65°C, adecuada para aplicaciones donde no se espere un calentamiento excesivo.
- **Acabado superficial:** la estructura presenta un acabado superficial de calidad, que puede ser mejorado mediante post-procesamiento, como lijado o aplicación de capas de protección.



Figura 14. Estructura del Robot

Fuente: extraído de <https://blog.bricogeek.com/noticias/robotica/brazo-robot-scara-que-puedes-imprimir-en-3d/>

B. Tornillería y sujeciones: la tornillería y las sujeciones son elementos clave para ensamblar y asegurar todas las partes de la estructura del robot de manera segura y estable.

Tornillos

- **M3x25, M3x30, M3x10, M8x45, M4x30 y M3x15:** cada uno diseñado para unir partes específicas de la estructura, con una longitud y grosor adecuados para su función.
- **Material:** acero inoxidable o acero galvanizado, que proporciona alta resistencia a la corrosión y durabilidad.
- **Cabeza:** tipo estrella y hexagonal, facilitando su ajuste con herramientas estándar y ofreciendo una sujeción firme.

Tuercas

- **Material:** acero inoxidable, complementando los tornillos y asegurando conexiones robustas.
- **Tipo:** hexagonales y de seguridad, que aseguran que las uniones no se aflojen debido a vibraciones o movimientos repetitivos.

Arandelas: incluyen arandelas planas y de presión para distribuir la carga de sujeción y evitar que los tornillos se aflojen.

Montaje: tornillos de varios tamaños y tipos, seleccionados específicamente para asegurar piezas como los motores, poleas y la estructura principal, garantizando que todas las partes permanezcan firmemente unidas durante la operación.



Figura 15. Tornillería

Fuente: extraído de <https://articulo.mercadolibre.com.co/MCO-854169650-kit-de-tornillos-hexagonal-de-acero-m3-m4-m5-accesorios-de- JM>

C. Elementos de transmisión de movimiento: estos componentes son responsables de transmitir el movimiento y la potencia desde los motores a las articulaciones del brazo robótico, asegurando un control preciso de la posición y la velocidad.

Correas dentadas

- **Longitudes:** correas de 160mm y 320mm con 80 y 160 dientes respectivamente, diseñadas para diferentes etapas del movimiento.
- **Material:** fabricadas en poliuretano con insertos de fibra de vidrio, lo que proporciona alta resistencia al desgaste, flexibilidad y estabilidad dimensional.
- **Ancho:** las correas suelen tener un ancho estándar (por ejemplo, 6 mm), optimizado para el tipo de carga y la relación de transmisión requerida.

- **Resistencia a la tracción:** alta resistencia a la tracción para evitar el estiramiento bajo carga, manteniendo la precisión del movimiento.

Poleas dentadas

- **Número de dientes:** 20 dientes, lo que ofrece una buena relación de transmisión y compatibilidad con las correas dentadas.
- **Material:** generalmente hechas de aluminio o acero, que proporcionan durabilidad y una relación peso/resistencia adecuada.
- **Diámetro del eje:** las poleas están diseñadas para ajustarse a ejes estándar, con diámetros típicos que van desde 5 mm a 8 mm.
- **Compatibilidad:** diseñadas para un ajuste preciso con las correas dentadas, minimizando el juego y asegurando una transmisión de potencia eficiente.

Acople flexible

- **Dimensiones:** acople de 5*8mm, utilizado para conectar el motor al eje de transmisión, absorbiendo desalineaciones menores y reduciendo vibraciones.
- **Material:** hecho de aluminio o acero inoxidable, lo que asegura flexibilidad sin comprometer la rigidez necesaria para la transmisión de movimiento.
- **Montaje:** fácil de instalar y ajustar, con tornillos de fijación que aseguran una conexión firme entre el motor y los ejes.

Estos elementos mecánicos son esenciales para garantizar la funcionalidad y precisión del brazo robótico, permitiendo movimientos coordinados y controlados en cada una de sus articulaciones



Figura 16. Correas dentadas

Fuente: extraído de <https://nk-technics.mx/correas-dentadas-de-poliuretano/perfil-t-metrico/correas-dentadas-pu-t25/correa-dentada-t2-5-sin-fin.htm>

5. Metodología

5.1 Tipo de proyecto

En el marco del proyecto de grado enfocado en el desarrollo de un kit robótico SCARA asequible y educativo para mejorar la enseñanza en regiones con recursos limitados, se adoptará un enfoque metodológico combinado, específicamente clasificado como "Aplicado y Exploratorio". Esta elección se fundamenta en la naturaleza del proyecto, que busca la creación de un producto tangible y práctico, el kit robótico SCARA, con aplicaciones directas en el ámbito educativo. Al mismo tiempo, el enfoque exploratorio se justifica por la necesidad de comprender a fondo las particularidades, restricciones y desafíos potenciales asociados con la implementación del kit en entornos específicos. Este enfoque no solo asegura la generación de un producto utilizable, sino que también permite la exploración de nuevas perspectivas y soluciones para mejorar la enseñanza de la robótica en contextos con recursos limitados, cumpliendo así con los objetivos del proyecto.

En este proyecto, se adoptará un enfoque mixto, combinando elementos cualitativos y cuantitativos en la recolección y análisis de datos. El aspecto cualitativo se enfocará en la recolección de datos sobre las percepciones, opiniones y experiencias de los usuarios (educadores y estudiantes) respecto al uso del kit robótico SCARA, particularmente en entornos educativos con recursos limitados. Esto permitirá comprender de manera profunda las necesidades y desafíos específicos que estos usuarios enfrentan.

Por otro lado, el enfoque cuantitativo se centrará en la medición objetiva del desempeño del kit robótico, como la precisión en los movimientos, el tiempo de respuesta, la durabilidad de los componentes, y el impacto cuantificable en el aprendizaje de los estudiantes. Estos datos permitirán evaluar la eficacia y la eficiencia del kit en un contexto educativo.

Por otro lado, el enfoque cuantitativo se centrará en la medición objetiva del desempeño del kit robótico, como la precisión en los movimientos, el tiempo de respuesta, la durabilidad de los componentes, y el impacto cuantificable en el aprendizaje de los estudiantes. Estos datos permitirán evaluar la eficacia y la eficiencia del kit en un contexto educativo.

Según el grado de manipulación de las variables, el proyecto se clasifica como no experimental, ya que no se realizarán manipulaciones deliberadas de variables para observar sus efectos, sino que se observarán y analizarán los fenómenos tal como ocurren en el entorno real.

Esto es acorde con la naturaleza exploratoria del proyecto, donde se busca comprender y describir cómo el kit robótico SCARA influye en el proceso educativo sin intervenir directamente en los factores que podrían afectar los resultados.

5.2 Método

El propósito fundamental de este proyecto es desarrollar entrenadores portátiles de robótica destinados a estudiantes de zonas rurales, programas de regionalización y otros actores en áreas remotas. Estos entrenadores tienen como objetivo respaldar los procesos de investigación formativa en las disciplinas asociadas a la robótica. El proyecto se llevará a cabo a través de una serie de acciones secuenciales, comenzando con el diseño del prototipo de un brazo robótico tipo SCARA mediante herramientas CAD. Posteriormente, se procederá al desarrollo del prototipo, seguido de pruebas de funcionamiento, que posteriormente permitan la replicación del mismo en forma de kits portátiles.

En la ejecución de este proyecto, se empleará el método de análisis-síntesis a nivel de hardware, lo que implica el diseño, la determinación de componentes y equipos necesarios, y su implementación. Es crucial destacar que todos estos pasos se llevarán a cabo considerando que los entrenadores portátiles deben ser elementos permanentes en el tiempo, permitiendo su funcionalidad itinerante. Esto implica su capacidad de desplazamiento y manipulación por parte de diversos actores involucrados en el proceso formativo en distintos municipios de las regiones especificadas. La metodología propuesta se ilustra en un diagrama secuencial que detalla las actividades planificadas para la ejecución del estudio.

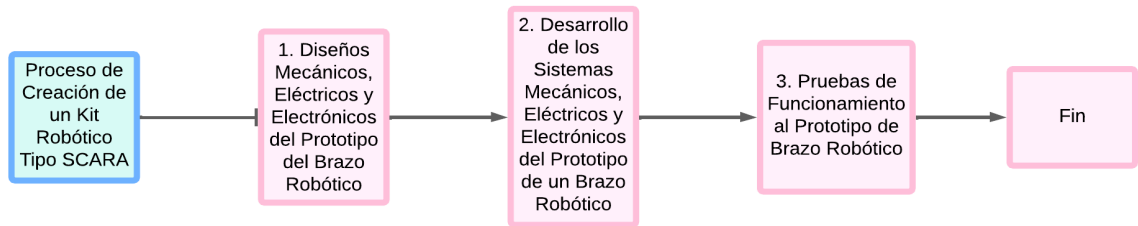


Figura 17. Metodología
Fuente: diseño propio

5.2.1 Diseños Mecánicos, Eléctricos y Electrónicos del Prototipo del Brazo Robótico.

En esta fase crítica, se llevará a cabo una exhaustiva exploración conceptual y técnica para la formulación de los diseños mecánicos, eléctricos y electrónicos del prototipo del brazo robótico. El diseño mecánico se enfocará en la creación de un manipulador tipo SCARA, teniendo en cuenta la ergonomía, la resistencia estructural y la eficiencia en su funcionamiento. Los análisis de simulación y modelado tridimensional mediante herramientas CAD permitirán optimizar la geometría y los materiales, garantizando la precisión y eficacia del manipulador.

En paralelo, el diseño eléctrico se centrará en la adaptación de la fuente de alimentación del sistema. Se analizarán detalladamente los requerimientos energéticos de los componentes eléctricos y electrónicos del brazo robótico. La selección de componentes, como motores y sensores, se realizará considerando la eficiencia energética y la compatibilidad con el diseño mecánico. Asimismo, se abordarán aspectos como la distribución de energía, la prevención de interferencias electromagnéticas y la seguridad eléctrica.

El diseño electrónico comprenderá la integración de sistemas de control y la interfaz hombre-máquina. Se determinarán los circuitos necesarios para el control de motores, la retroalimentación de sensores y la comunicación entre componentes electrónicos. La elección de plataformas de control, como microcontroladores o microprocesadores, se basará en criterios de eficiencia y versatilidad para adaptarse a diversas aplicaciones educativas.

El resultado de esta fase será un conjunto integral de diseños que proporcionarán las bases sólidas para la construcción del prototipo del entrenador portátil de robótica, asegurando una alineación coherente entre los aspectos mecánicos, eléctricos y electrónicos del brazo robótico.

5.2.2 Desarrollo de los Sistemas Mecánicos, Eléctricos y Electrónicos del Prototipo de un Brazo Robótico. Tras la elaboración detallada de los diseños mecánicos, eléctricos y electrónicos, se procederá a la fase crucial de construcción del prototipo del entrenador portátil de robótica. Este proceso implica la materialización de los conceptos teóricos plasmados en los diseños, y se realizará en espacios de laboratorio institucionales o directamente en las regiones, garantizando la accesibilidad a las herramientas y equipos necesarios para la ejecución del proyecto.

Aspectos Mecánicos: la construcción del prototipo mecánico incluirá la fabricación de las piezas estructurales del manipulador tipo SCARA, siguiendo las especificaciones de los diseños. Se emplearán técnicas de manufactura precisa, como corte láser o impresión 3D, para asegurar la fidelidad del prototipo al diseño original. La atención se centrará en la calidad de los materiales y la precisión en la construcción para garantizar un rendimiento mecánico óptimo.

Aspectos Eléctricos y Electrónicos: simultáneamente, se avanzará en la implementación de los sistemas eléctricos y electrónicos. Esto incluirá la instalación de la fuente de alimentación, la conexión de motores y sensores, y la disposición de circuitos electrónicos en la estructura diseñada. Se prestará especial atención a la organización del cableado para evitar interferencias y asegurar una conexión eléctrica estable. La programación de los microcontroladores o microprocesadores se realizará para coordinar los movimientos del brazo robótico y permitir su control.

Integración de Sistemas: la fase de desarrollo implicará la integración progresiva de los sistemas mecánicos, eléctricos y electrónicos. La colaboración estrecha entre ingenieros mecánicos y eléctricos garantizará la cohesión de los elementos y la resolución inmediata de posibles desafíos que surjan durante la implementación. Esta etapa es crucial para asegurar que el prototipo funcione de manera sinérgica y cumpla con los requisitos de diseño.

Esta fase de construcción constituye un paso fundamental en el proyecto, llevando los diseños a la realidad y proporcionando la base tangible para las siguientes etapas de pruebas y replicación del prototipo.

5.2.3 Pruebas de Funcionamiento al Prototipo de Brazo Robótico. Una vez completada la construcción del prototipo del entrenador portátil de robótica, se procederá a una serie detallada de pruebas de funcionamiento. Estas pruebas están diseñadas para evaluar la eficacia y eficiencia del brazo robótico en condiciones simuladas de aplicación. Cada componente mecánico, eléctrico y electrónico será sometido a escenarios específicos para verificar su desempeño e identificar posibles áreas de mejora.

Pruebas Mecánicas: en esta etapa, se evaluará la resistencia estructural del brazo robótico. Se realizarán pruebas de carga para determinar su capacidad para manipular objetos de diferentes pesos y tamaños. Además, se verificará la precisión en la ejecución de movimientos y la repetibilidad de las trayectorias planificadas. La velocidad y la suavidad en los desplazamientos también serán aspectos clave a evaluar.

Pruebas Eléctricas y Electrónicas: las pruebas eléctricas se centrarán en la estabilidad y fiabilidad de la fuente de alimentación, asegurando que proporcione la energía necesaria de manera constante. Se verificará la respuesta de los motores y la precisión en la lectura de los sensores. Las pruebas electrónicas evaluarán la programación de los microcontroladores, garantizando que el brazo responda de manera adecuada a los comandos y que la comunicación entre los componentes sea fluida.

Pruebas Integradas: las pruebas integradas serán cruciales para evaluar el rendimiento del sistema como un todo. Se simularán escenarios realistas de aplicación donde el brazo robótico deberá ejecutar tareas específicas. Esto incluirá la coordinación de movimientos, la interacción con objetos y la capacidad de respuesta a comandos externos. Se prestará especial atención a la sincronización entre los sistemas mecánicos y electrónicos para garantizar un funcionamiento coherente y eficiente.

Identificación y Corrección de Problemas: durante las pruebas, se documentarán detalladamente cualquier problema o anomalía identificada. Este proceso permitirá una retroalimentación rápida para corregir posibles fallos o realizar ajustes necesarios en el diseño o

la implementación. La iteración y mejora continua son fundamentales para asegurar que el prototipo cumpla con los estándares de calidad y desempeño esperados.

Validación de Aplicaciones Prácticas: además de las pruebas técnicas, se llevarán a cabo proyectos prácticos que involucren la aplicación del brazo robótico en situaciones del mundo real. Esto proporcionará información valiosa sobre la utilidad y versatilidad del entrenador portátil en contextos educativos y de investigación. Estos proyectos prácticos serán esenciales para validar la efectividad del prototipo en la enseñanza de conceptos robóticos y su aplicación en escenarios prácticos.

5.3 Instrumentos de recolección de información

5.3.1 Fuentes primarias. Documentos que proporcionan especificaciones técnicas y estructurales originales del brazo robótico, obtenidos directamente del proceso de diseño y documentación detallada que refleja los resultados y rendimiento durante las pruebas prácticas del prototipo, constituyendo información original del proceso. También se usarán datos provenientes de registros de proveedores y costos asociados, fundamentales para la replicación del prototipo. Finalmente se contará con comunicación directa con los profesores asesores y diseñadores del brazo robótico, quienes ofrecieron información de primera mano basada en su experiencia y conocimiento.

5.3.2 Fuentes secundarias. Información interpretativa proporcionada por profesores asesores, que complementa la información original y directa obtenida durante el proceso de diseño y construcción, además de la consulta de libros, artículos académicos y revistas científicas relacionadas con la robótica educativa y el diseño de brazos robóticos SCARA, que respaldan y enriquecen la comprensión del contexto teórico y que son listadas en la bibliografía.

6. Resultados

6.1 Diseño del prototipo de brazo robótico “SCARA”

El diseño del prototipo del brazo robótico tipo SCARA se llevó a cabo utilizando una combinación de herramientas CAD avanzadas y simulaciones electrónicas para asegurar la precisión y funcionalidad. Este proceso resultó fundamental para alcanzar los objetivos tanto en el diseño mecánico como en el control electrónico del brazo.

6.1.1 Modelación de las piezas en CAD. El diseño de las piezas mecánicas se realizó mediante Autodesk Inventor y Fusion 360, tomando como base un modelo referenciado de un curso de UdeMY sobre robots SCARA. A partir de este modelo inicial, se hicieron varias modificaciones para ajustar las dimensiones según las necesidades del proyecto. Las modificaciones incluyeron el redimensionamiento de los eslabones del brazo, los soportes de los motores y otros componentes críticos para asegurar su correcto funcionamiento dentro de los límites establecidos. Además, se modelaron las piezas mecánicas para su impresión en 3D, y las piezas electrónicas, junto con los tornillos, varillas y motores, fueron exportadas al software para poder integrar un modelo completo del sistema. A continuación, se presenta la descripción de cada pieza, junto con su imagen de modelación correspondiente:

Varilla lisa. Esta varilla actúa como eje guía para el movimiento lineal de los componentes del sistema, su función es mantener el movimiento lineal fluido entre las piezas móviles.

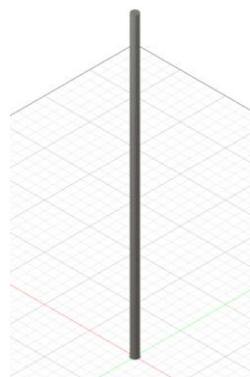


Figura 18. Varilla lisa
Fuente: diseño propio

Tornillo sin fin. Es un componente clave que convierte el movimiento rotatorio en movimiento lineal, su función es facilitar la transmisión de movimiento rotatorio, actuando como el mecanismo principal del brazo robótico.

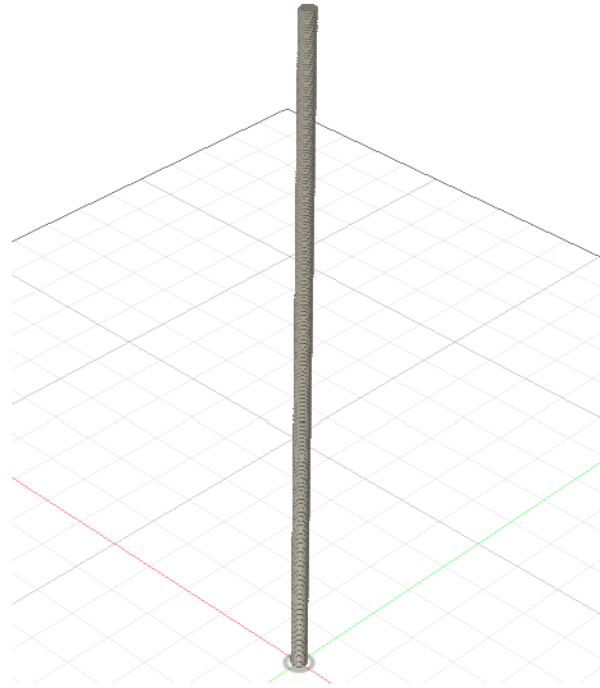


Figura 19. Tornillo sin fin
Fuente: diseño propio

Tornillo de ajuste. Tornillo diseñado para ajustar y asegurar la posición de los componentes en el montaje, permite garantizar la estabilidad y precisión en la fijación de las piezas.

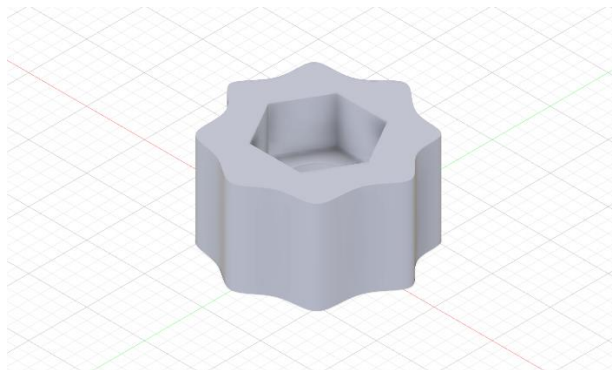


Figura 20. Tornillo de ajuste
Fuente: diseño propio

Spacers. Conjunto de espaciadores diseñados para mantener la distancia adecuada entre los componentes del brazo robótico. Incluyen varios tamaños como de 5.5 mm y 2 mm, que evitan interferencias y colisiones, asegurando un ensamblaje correcto y un movimiento eficiente de las partes móviles, su función consiste en proveer el espacio necesario para el movimiento libre y fluido entre componentes.

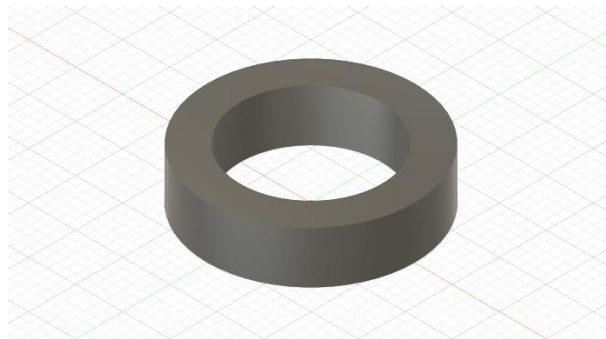


Figura 21. Spacers
Fuente: diseño propio

Soporte de mesa. Estructura que sirve de base para el montaje del brazo robótico, su función consiste en proporcionar estabilidad y soporte en la configuración del sistema.

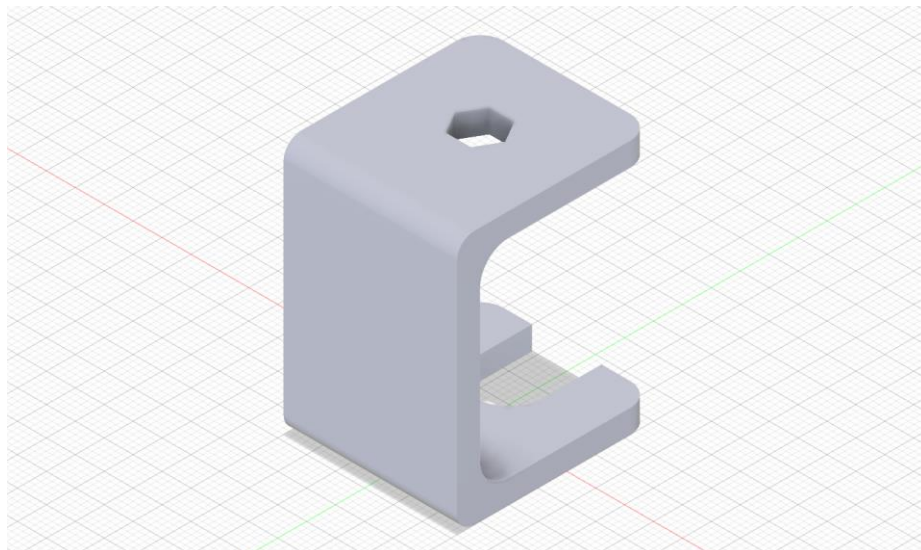


Figura 22. Soporte
Fuente: diseño propio

Rodamiento pequeño v1. Diseñado para permitir el movimiento rotacional en componentes pequeños, permite facilitar un giro suave y controlado en las partes móviles.

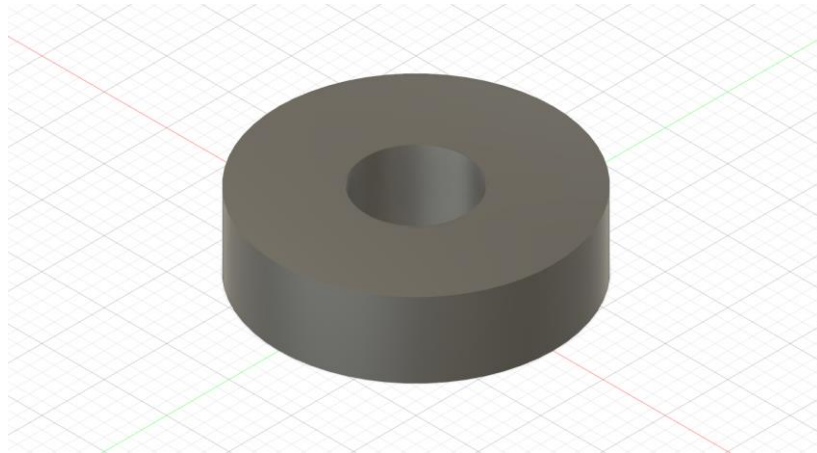


Figura 23. Rodamiento pequeño
Fuente: diseño propio

Rodamiento lineal. Permite el movimiento lineal a través de su diseño para asegurar un desplazamiento fluido y sin fricción en el brazo robótico.

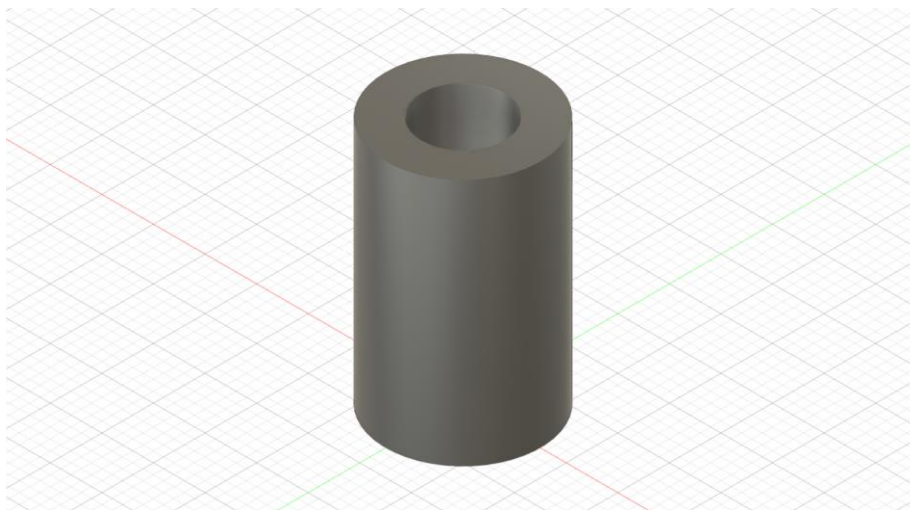


Figura 24. Rodamiento lineal
Fuente: diseño propio

Rodamiento Estándar. Este rodamiento estándar permite el movimiento rotacional en el sistema para mejorar la eficiencia y suavidad en las rotaciones de las piezas.

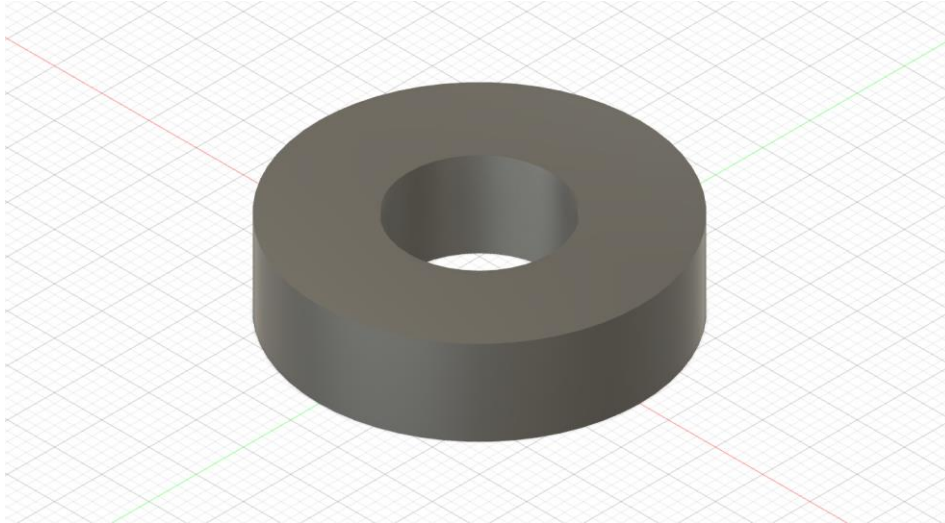


Figura 25. Rodamiento estándar
Fuente: diseño propio

Polea. Componente que transmite fuerza a través de un sistema de poleas para facilitar el movimiento de transmisión dentro del brazo robótico.

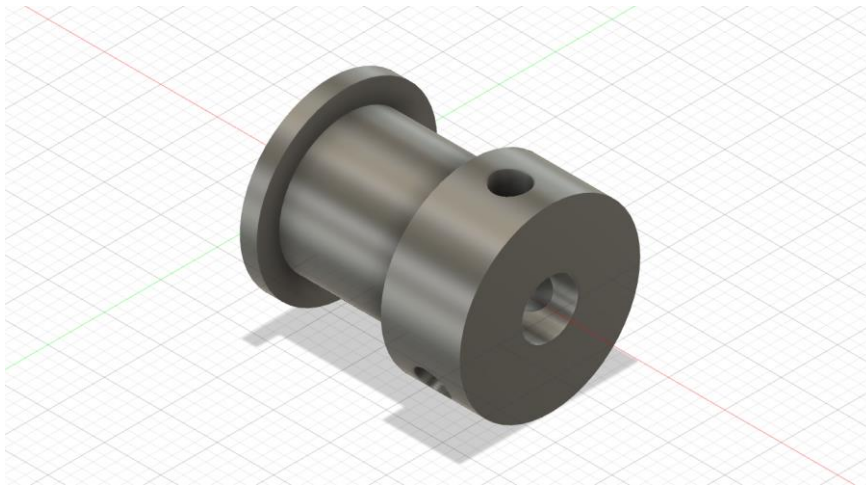


Figura 26. Polea
Fuente: diseño propio

Pinza. Herramienta de agarre que permite manipular objetos, actúa en el extremo del brazo robótico para ejecutar tareas de sujeción.

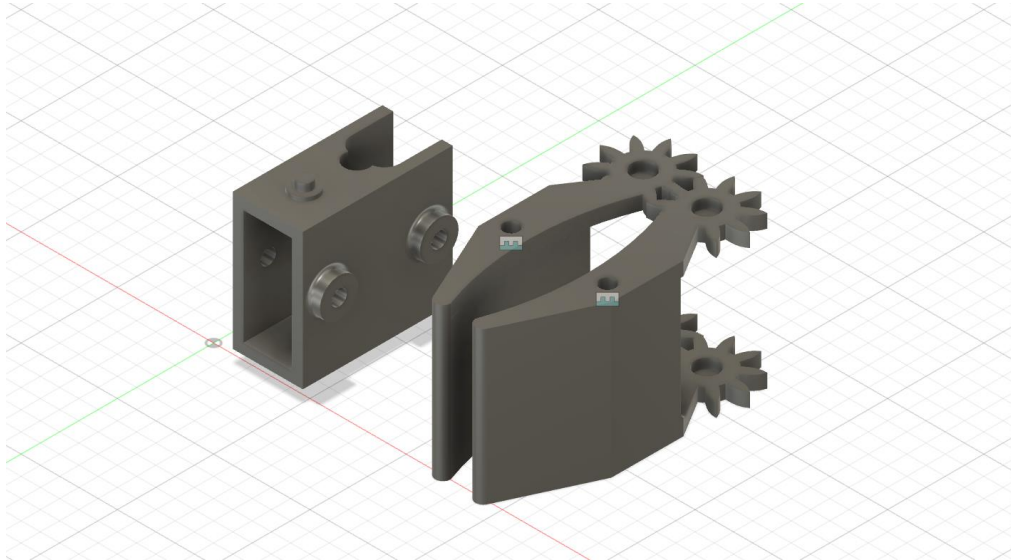


Figura 27. Pinza
Fuente: diseño propio

Conector de articulación. Pieza que une las articulaciones del brazo, permitiendo el movimiento y facilita el movimiento rotacional en las conexiones del brazo.

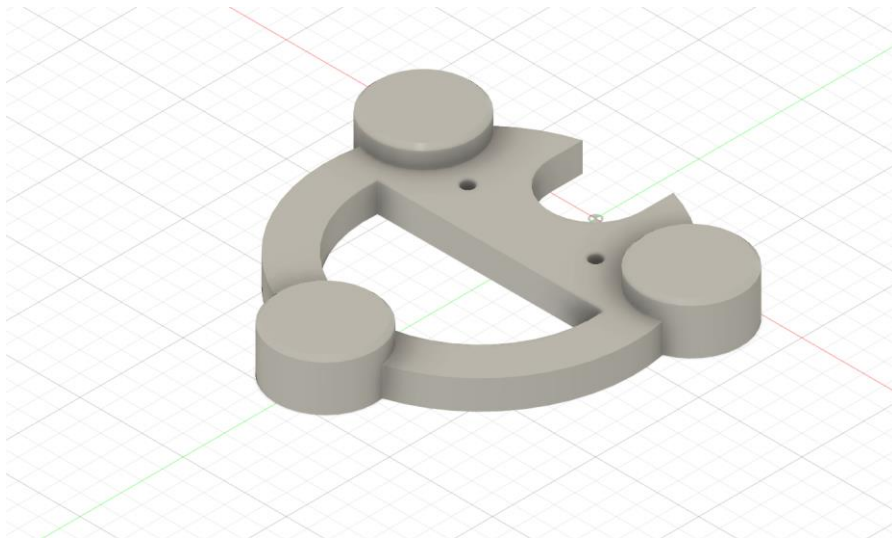


Figura 28. Conector
Fuente: diseño propio

Adaptador de base a eje rotatorio. Conector que distribuye fuerzas de rotación desde la base del sistema, permite asegurar la conexión entre la base y los componentes móviles.



Figura 29. Adaptador
Fuente: diseño propio

Conector de transmisión a componentes móviles v1. Parte que une el sistema de transmisión a las piezas móviles del brazo, permite asegurar una transferencia eficiente de movimiento.

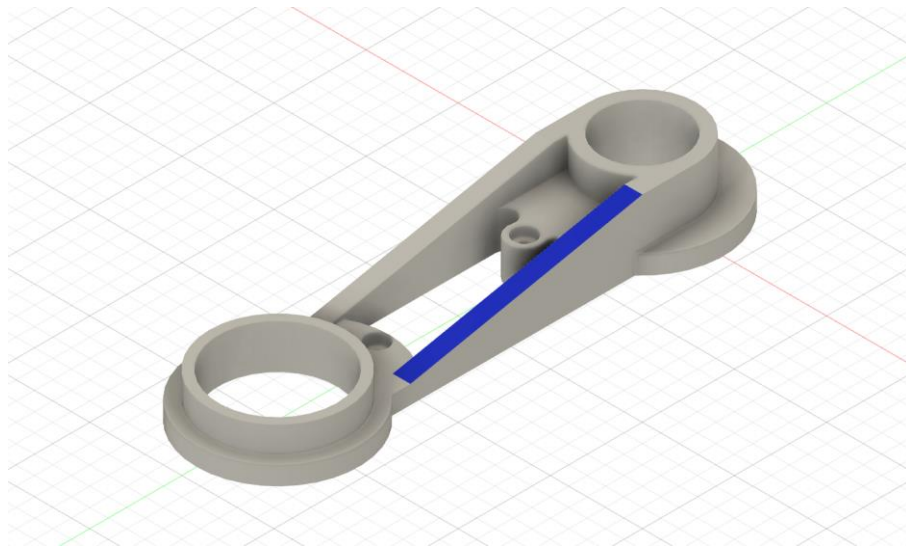


Figura 30. Conector transmisión
Fuente: diseño propio

Conector estructural secundario. Pieza que mejora la estabilidad estructural uniendo componentes secundarios, su función consiste en aumentar la rigidez del brazo robótico.

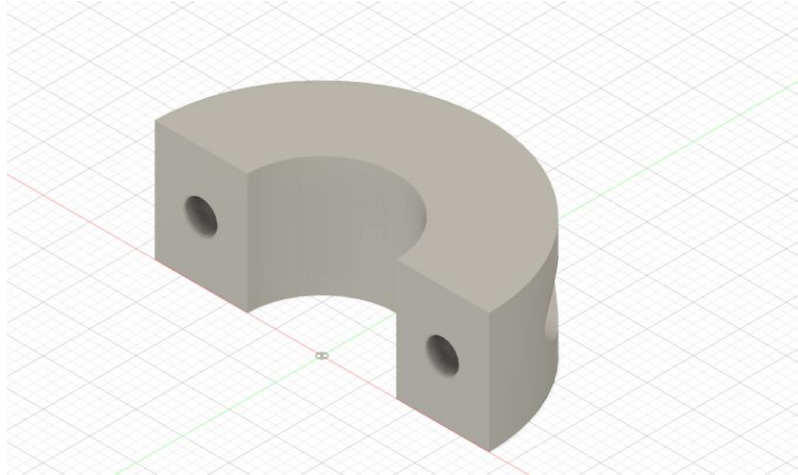


Figura 31. Conector estructural
Fuente: diseño propio

Unión pequeña de partes móviles. Componente que asegura la conexión entre piezas móviles, busca proveer estabilidad y seguridad en las articulaciones del brazo.

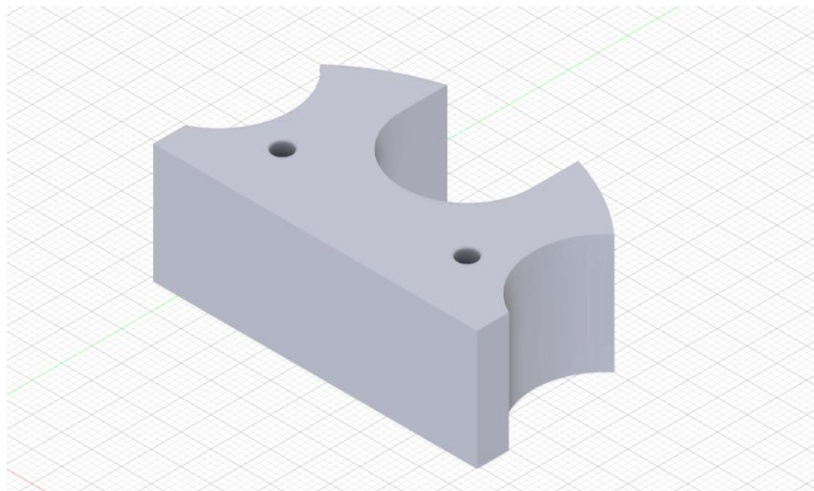


Figura 32. Unión
Fuente: diseño propio

Base estructural principal. Soporte que conecta y estabiliza todas las partes del brazo robótico, su función consiste en proporcionar una base sólida para el sistema completo.

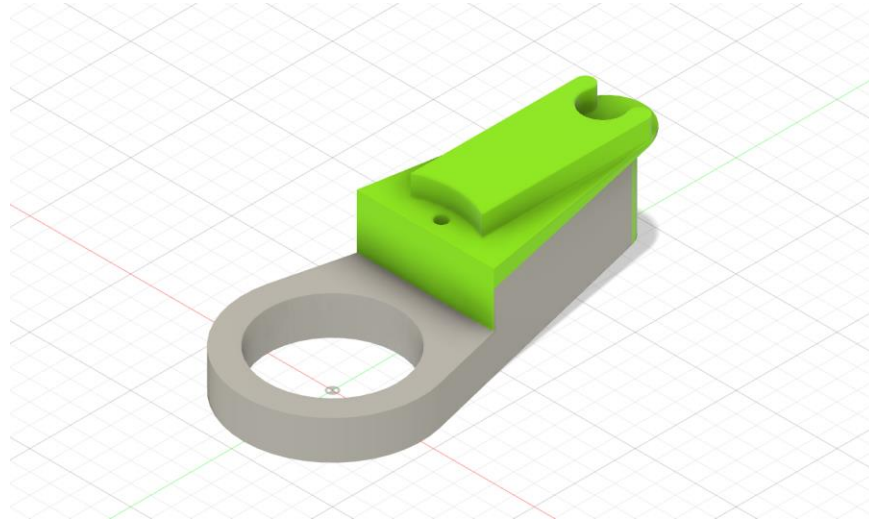


Figura 33. Base
Fuente: diseño propio

Soporte de alineación. Componente que mantiene la alineación de las partes durante el movimiento, su función consiste en asegurar que el sistema opere de manera eficiente y coordinada.

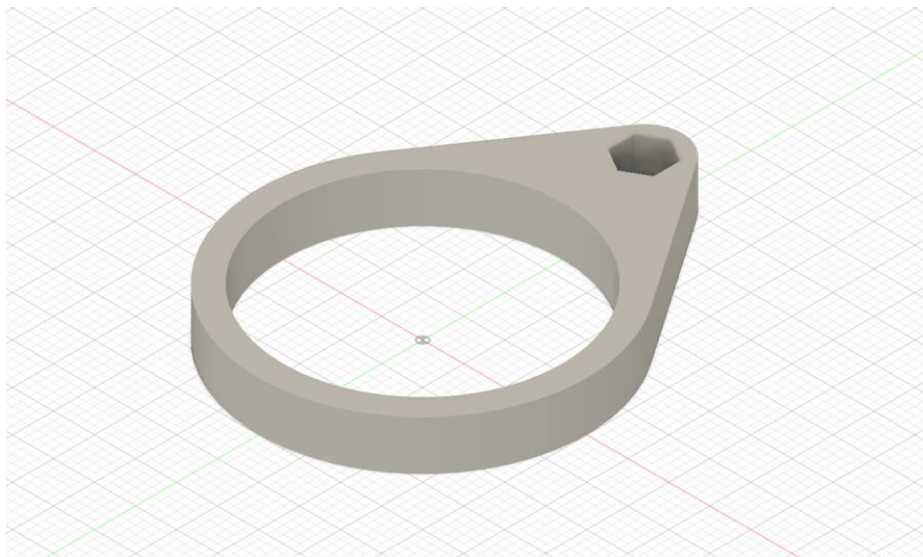


Figura 34. Soporte
Fuente: diseño propio

Conector de actuador a brazo. Conector que transmite el movimiento del actuador al brazo robótico, permite asegurar que el brazo se mueva correctamente en respuesta a los comandos del actuador.

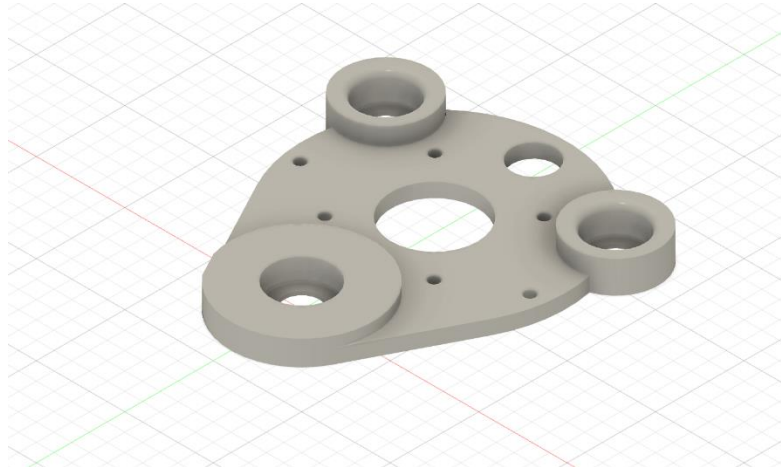


Figura 35. Conector actuador
Fuente: diseño propio

Conector de soporte en transmisión. Pieza que asegura la rigidez del sistema de transmisión para mantener la posición y estabilidad de los componentes de transmisión.

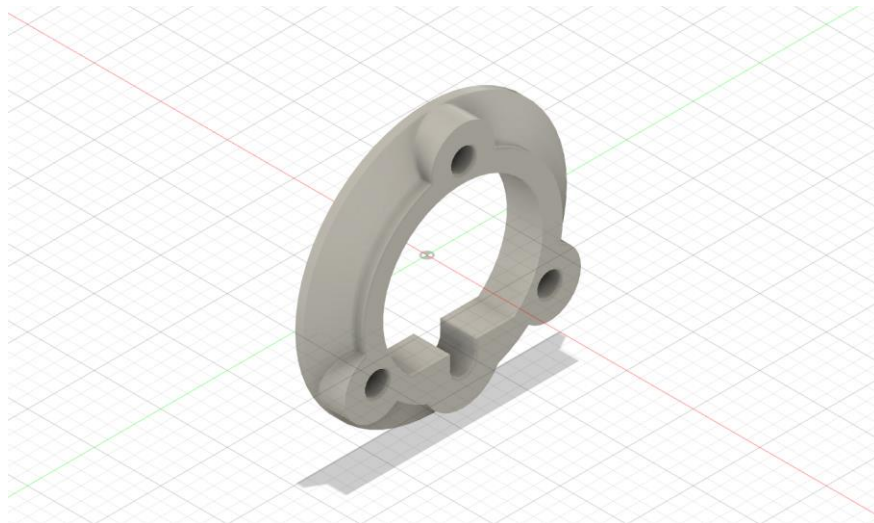


Figura 36. Conector soporte
Fuente: diseño propio

Nema 17 42HS48-1684 v1. Motor paso a paso que proporciona un control preciso del movimiento, permite movimientos exactos y programados del brazo robótico.

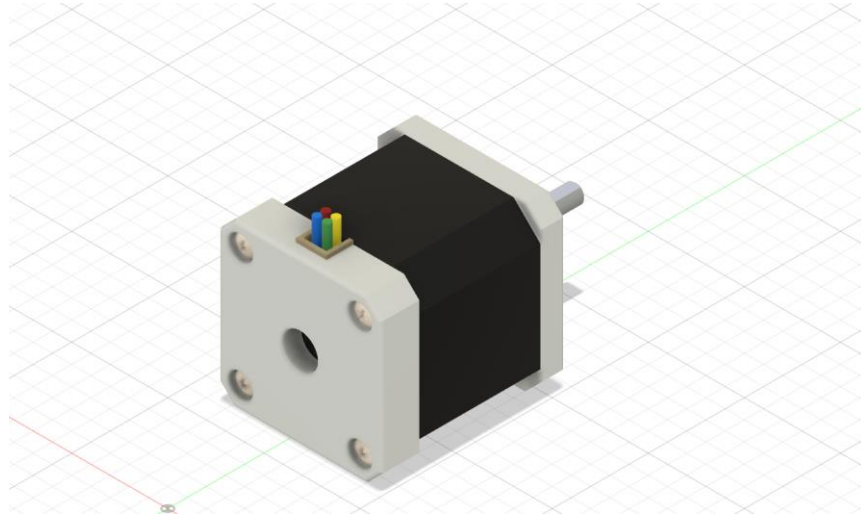


Figura 37. Nema 17

Fuente: diseño propio

Motor Servo MG90S. Motor servo diseñado para actuar como parte del mecanismo de la pinza, su función consiste en proporcionar movimiento controlado en el sistema robótico.

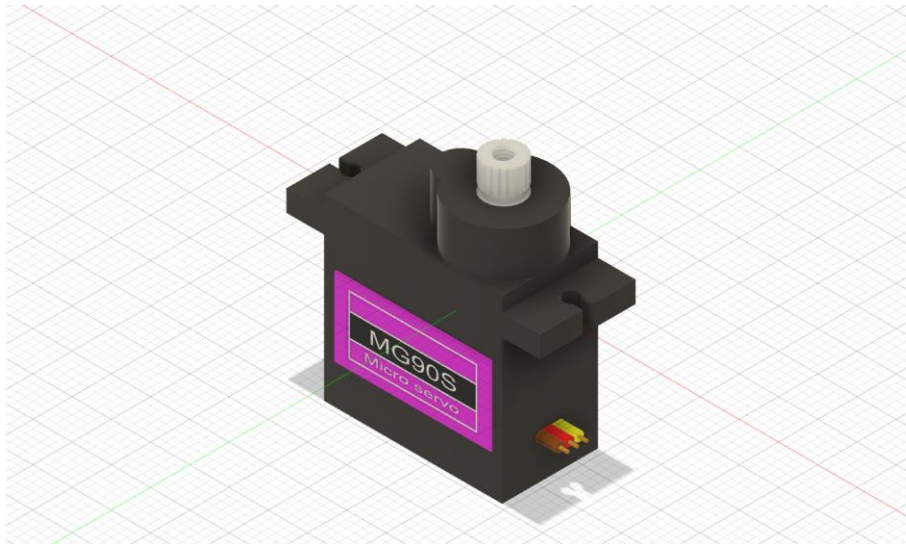


Figura 38. Motor servo

Fuente: diseño propio

Eje de correa. Componente que integra el sistema de transmisión de correas, su función consiste en mantener la alineación y tensión adecuada en el sistema de transmisión.

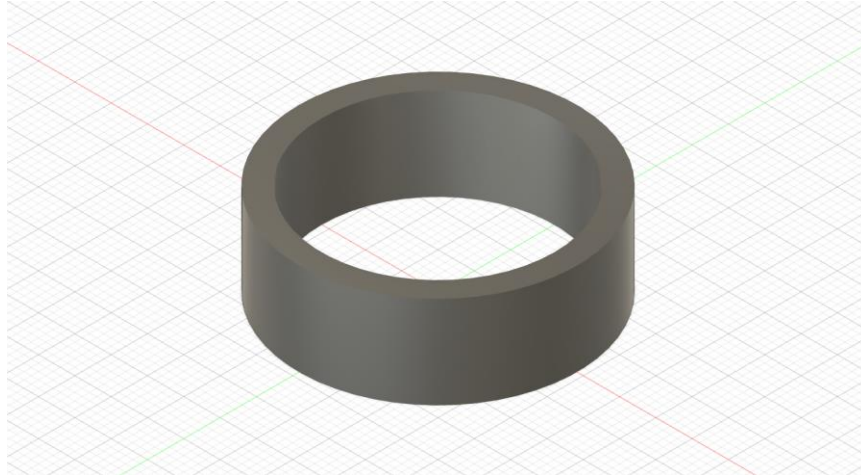


Figura 39. Eje de correa
Fuente: diseño propio

Tornillo de ajuste para tornillo sin fin v3. Tornillo que permite el ajuste del tornillo sin fin para una alineación perfecta, permite asegurar un funcionamiento eficiente y preciso del mecanismo.

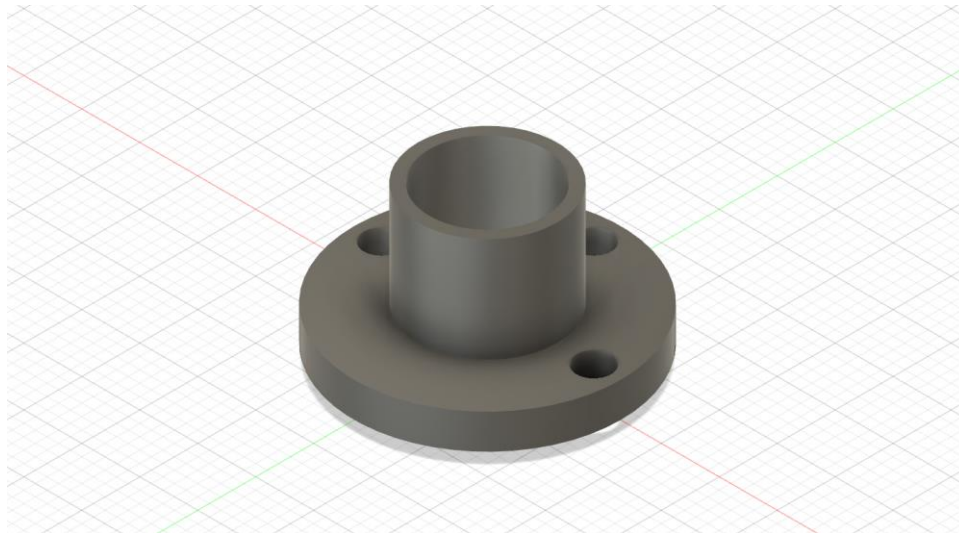


Figura 40. Tornillo de ajuste
Fuente: diseño propio

Correa Pequeña. Correa utilizada en la transmisión de movimiento entre componentes, facilita la transferencia de movimiento y energía en el sistema.

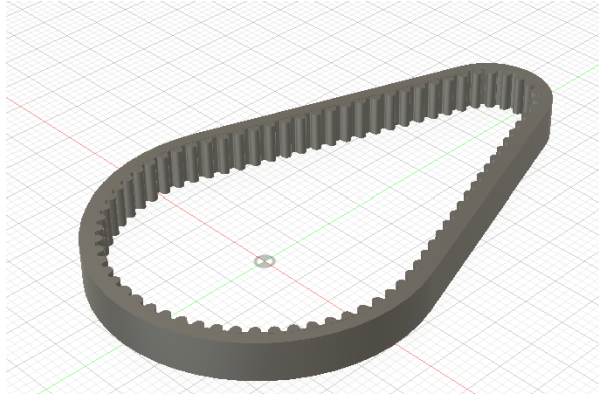


Figura 41. Correa pequeña
Fuente: diseño propio

Correa Grande. Correa principal que conecta varios componentes del brazo robótico, proporciona una transmisión eficiente de movimiento en el sistema.

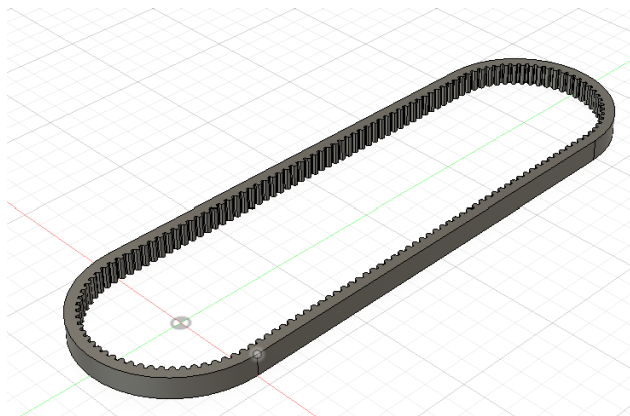


Figura 42. Correa grande
Fuente: diseño propio

Caja servo. Alojamiento diseñado para proteger el motor servo, permite asegurar el motor y prevenir daños durante el funcionamiento.

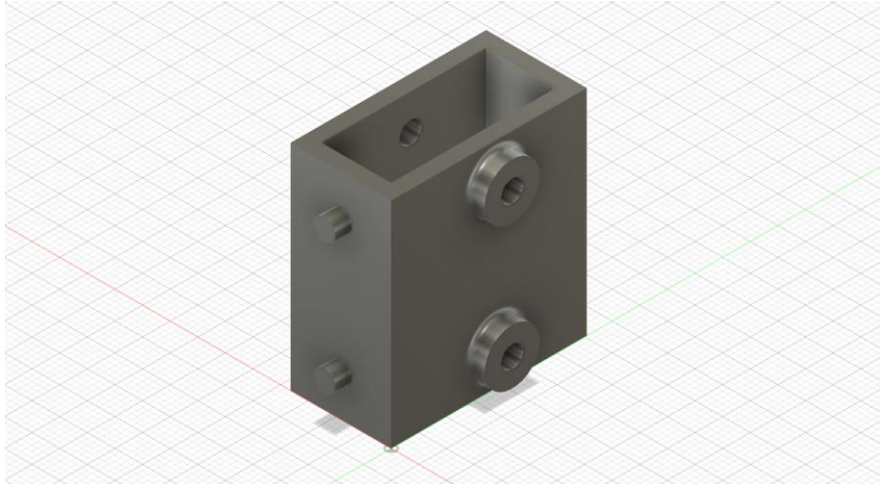


Figura 43. Caja servo
Fuente: diseño propio

Brazo extendido izquierdo. Componente que amplía el alcance del brazo en el lado izquierdo de la pinza, su función es aumentar la capacidad de movimiento y funcionalidad del brazo robótico.

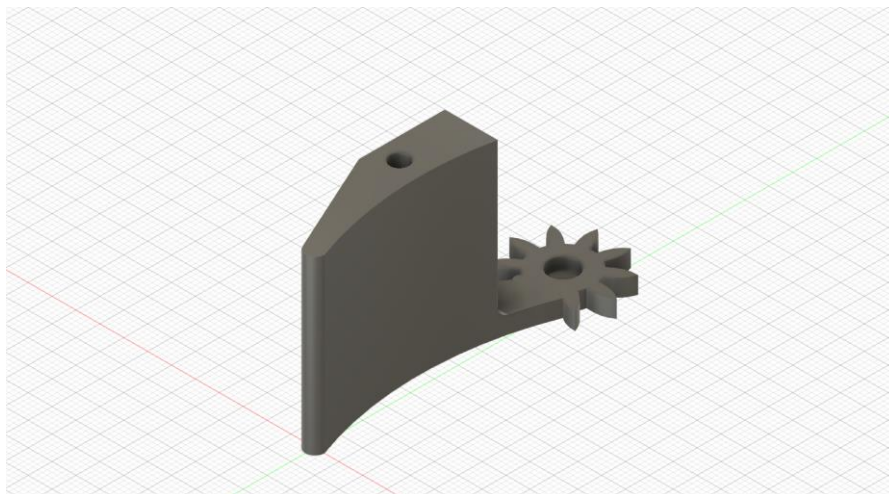


Figura 44. Brazo extendido izquierdo
Fuente: diseño propio

Brazo extendido derecho. Componente que amplía el alcance del brazo en el lado derecho de la pinza, su función consiste en proporcionar mayor capacidad de manipulación del brazo robótico.

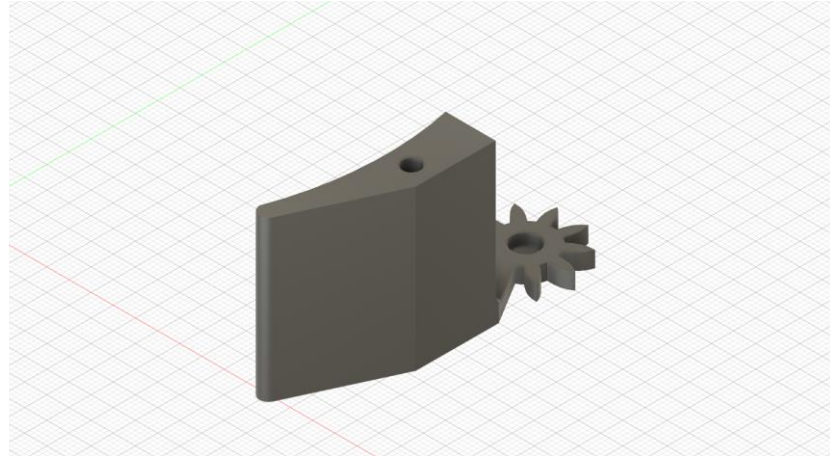


Figura 45. Brazo extendido izquierdo
Fuente: diseño propio

Brazo derecho. Soporte principal para la articulación derecha de la pinza, permite un movimiento controlado en la parte derecha del sistema.

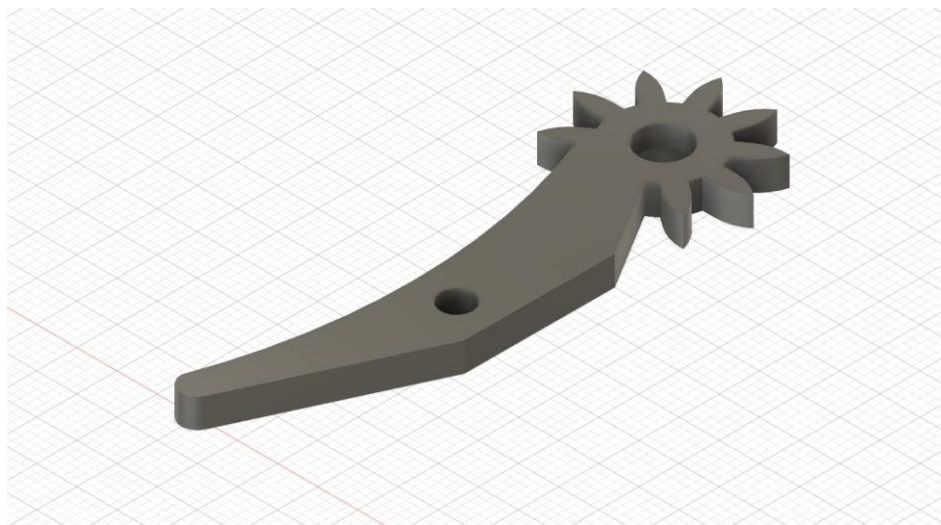


Figura 46. Brazo derecho
Fuente: diseño propio

Brazo izquierdo. Soporte para la articulación izquierda de la pinza, facilita el movimiento en la parte izquierda del sistema.

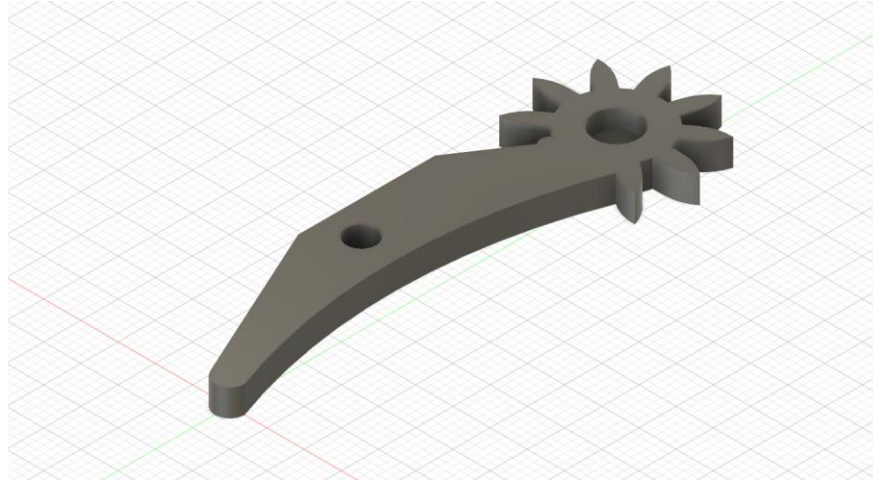


Figura 47. Brazo izquierdo
Fuente: diseño propio

Amarre. Pieza que asegura la posición de los componentes del brazo robótico, con el fin de mantener la integridad estructural del sistema durante la operación.

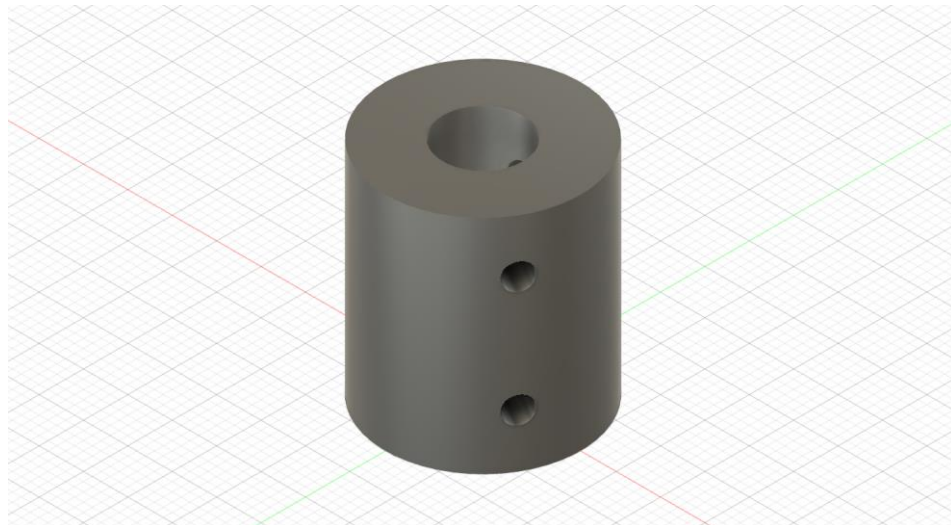


Figura 48. Amarre
Fuente: diseño propio

Soporte de pinza. Componente donde se ensambla la pinza, permitiendo su movimiento, conecta la pinza al brazo robótico, permitiendo su manipulación de objetos.

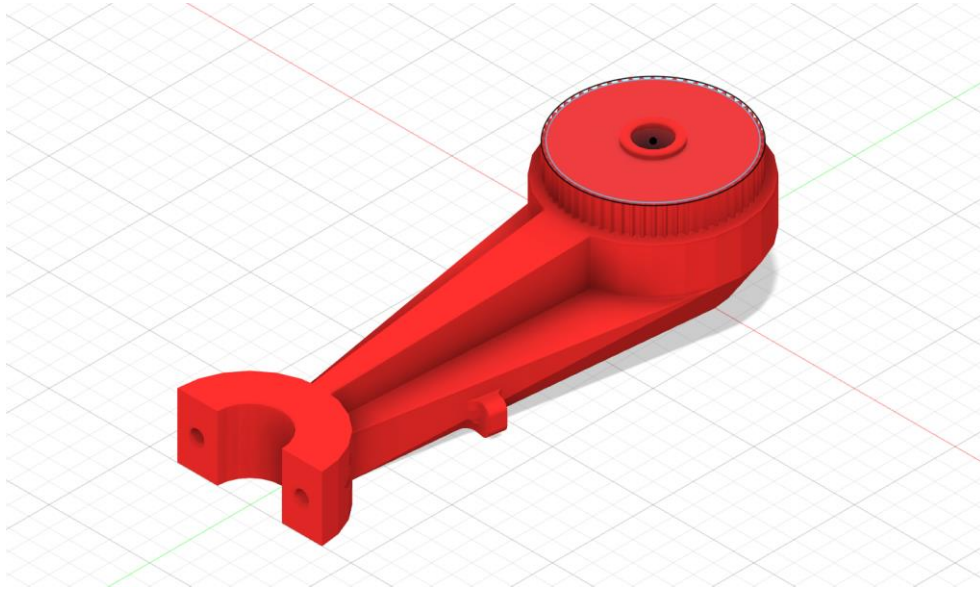


Figura 49. Soporte pinza
Fuente: diseño propio

El uso de estas herramientas CAD permitió realizar simulaciones del ensamblaje completo de todas las anteriores piezas, garantizando que todos los componentes encajaran de manera adecuada y que los ejes de movimiento fueran óptimos para los motores paso a paso seleccionados. Estas simulaciones fueron esenciales para prevenir problemas de montaje y optimizar el espacio en el diseño final. A continuación, se presenta una imagen de la simulación del brazo robótico ensamblado en Fusion 360, que refleja la integridad y funcionalidad del diseño propuesto.

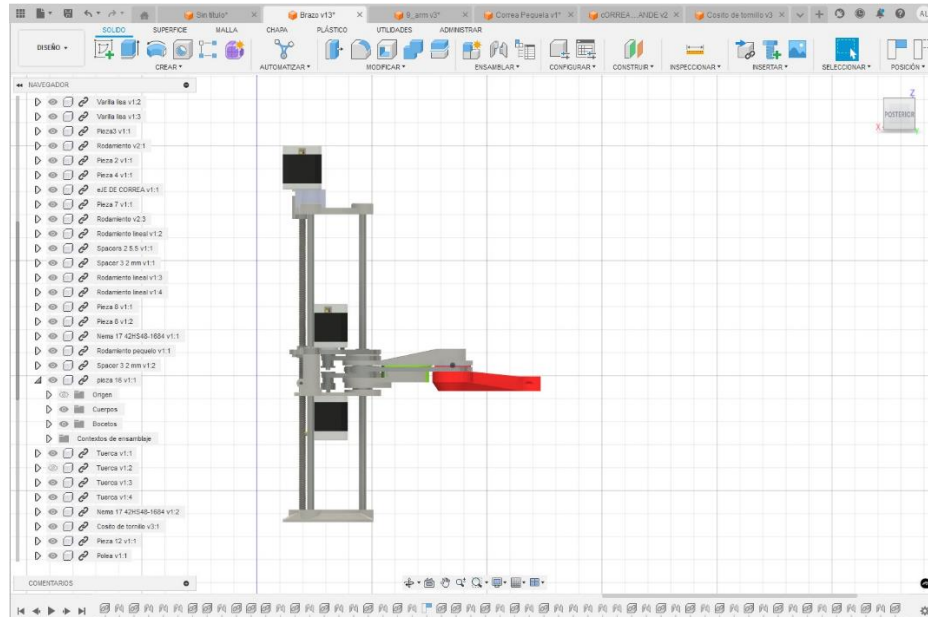


Figura 50. Simulación del ensamblaje

Fuente: diseño propio

6.1.2 Diseño de las correas. Un desafío particular en el diseño fue la fabricación de las correas de transmisión. Después de una búsqueda exhaustiva en el mercado, se concluyó que las correas comerciales no cumplían con las especificaciones requeridas para el prototipo. Por esta razón, se decidió modelar las correas desde cero utilizando CAD, para luego imprimirlas en TPU (poliuretano termoplástico), un material que proporciona la flexibilidad y resistencia necesarias para soportar las tensiones durante la operación del brazo.

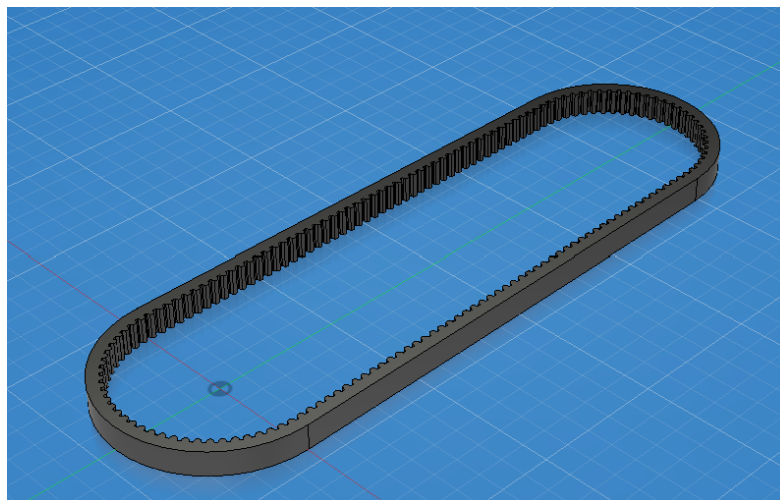


Figura 51. Modelación de correas

Fuente: diseño propio

El uso de TPU en la impresión de las correas permitió asegurar que estas tuvieran la elasticidad y durabilidad adecuadas, garantizando un movimiento suave y preciso entre los motores y los eslabones del brazo. Por otro lado, las poleas requeridas fueron adquiridas comercialmente y se ajustaron perfectamente a las correas diseñadas, logrando un sistema de transmisión robusto y eficiente.

6.1.3 Simulación y validación del circuito electrónico en Tinkercad. Para asegurar el correcto funcionamiento del sistema de control del brazo robótico, se realizó la simulación del circuito electrónico utilizando la plataforma Tinkercad. Esta herramienta permitió recrear virtualmente todas las conexiones electrónicas, incluyendo la placa Arduino Uno, sin embargo, debido a las limitaciones de Tinkercad, el motor NEMA17 y los drivers A4988 no están disponibles en esta plataforma. Para la simulación, se utilizó una configuración alternativa con servomotores, lo que permitió validar que el código funcionaba al enviar los ángulos de rotación a cuatro motores a través del monitor serial.

Durante la simulación, se probaron las secuencias de control y se verificó que las señales enviadas desde el Arduino fueran correctamente interpretadas. Los servomotores permitieron replicar los desplazamientos necesarios en los ejes X, Y y Z del brazo SCARA, comprobando la lógica de funcionamiento del código. La simulación también permitió evaluar el flujo de corriente y la distribución de voltaje entre los componentes, asegurando que el circuito fuera estable y adecuado para la operación continua del robot en un entorno virtual. Esta simulación previa fue clave para validar el sistema de control antes de llevarlo al entorno real con los componentes finales.

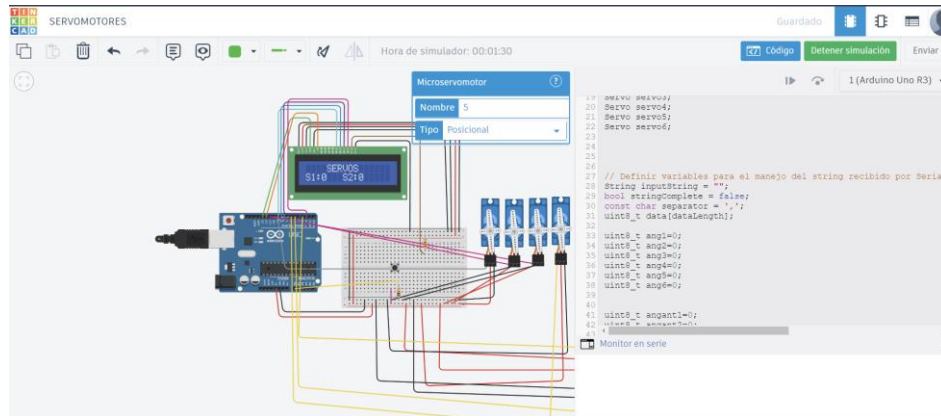


Figura 52. Simulación en Tinkercad

Fuente: disponible en

https://drive.google.com/file/d/1w0BhRFp5_dYov1IbDvrxEqynOmnZYdjY/view?usp=sharing

Las pruebas realizadas en la simulación confirmaron que el circuito era funcional y que el sistema de control podía manejar el movimiento del brazo con precisión. Esto fue un paso crucial antes de pasar a la implementación física del sistema, ya que permitió realizar ajustes en el código y la configuración del hardware sin necesidad de ensamblar los componentes reales.

6.2 Desarrollo del prototipo de brazo robótico “SCARA”

Después de completar el diseño y simulación de los mecanismos del brazo robótico "SCARA", se inició el desarrollo físico del prototipo a través de la impresión 3D de las piezas y componentes necesarios. Este proceso fue fundamental para transformar las ideas en una estructura tangible, permitiendo avanzar hacia la fase de ensamblaje y pruebas funcionales. La selección de materiales y parámetros de impresión fue muy importante para asegurar la durabilidad y eficiencia del prototipo en su funcionamiento.

6.2.1 Impresión 3D de piezas y correas. El desarrollo del prototipo del brazo robótico "SCARA" comenzó con la impresión de sus componentes fundamentales utilizando tecnología de impresión 3D. Se seleccionaron dos tipos de materiales según la función de las piezas:

Tabla 1.

Parámetros de impresión de piezas estructurales y correas flexibles

Características	Piezas Estructurales (PLA)	Correas Flexibles (TPU)
Material	PLA (ácido poliláctico)	TPU (poliuretano termoplástico)
Altura de Capa	0.2 mm	0.2 mm
Grosor de la Pared	0.84 mm	1 mm
Densidad de Relleno	15%	20%
Temperatura de Impresión	210 °C	230 °C
Velocidad de Impresión	30 mm/s	25 mm/s

Fuente: elaboración propia

A partir de la anterior tabla se procedió con la parametrización del proceso de impresión de la correa en el software Ultimaker Cura, como se puede apreciar en la siguiente figura.

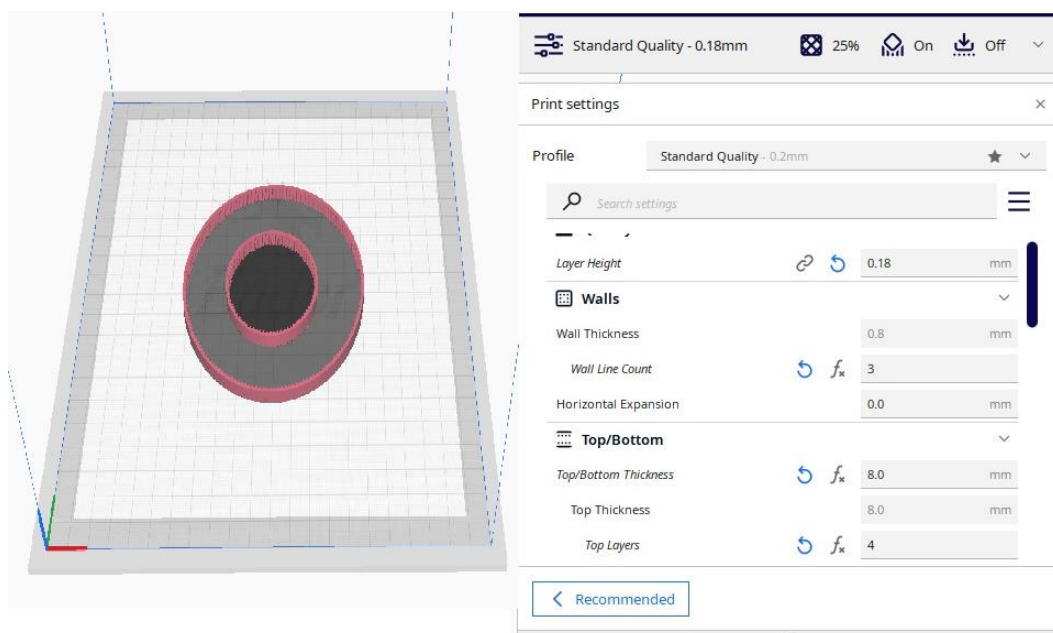


Figura 53. Parámetros de impresión para correas

Fuente: diseño propio

Para el ensamblaje final de este proyecto, se imprimieron en 3D varias piezas clave, utilizando diferentes materiales y configuraciones específicas según la función de cada componente. Las piezas estructurales, que requerían mayor rigidez y durabilidad, se imprimieron en PLA, un material termoplástico fácil de manipular y confiable para estructuras sólidas. Estas piezas se imprimieron con una altura de capa de 0.2 mm, a una velocidad de 50

mm/s, con la cama caliente a 60 °C y la boquilla a 200 °C, para asegurar una buena adherencia entre capas y un acabado de calidad.

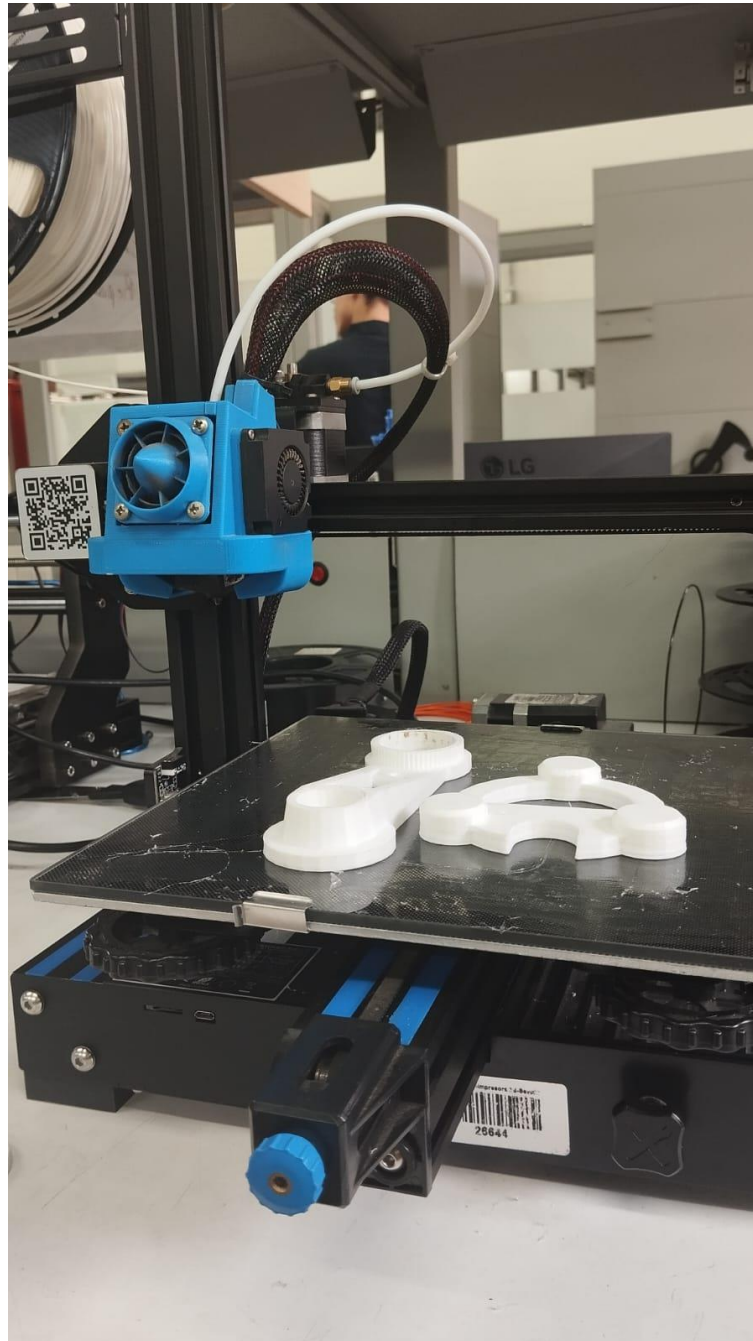


Figura 54. Piezas impresas en PLA sobre la cama de impresión
Fuente: diseño propio

Algunas piezas adicionales, necesarias para el ensamblaje de componentes específicos, también se imprimieron en PLA. Estas se configuraron con un relleno del 20% en patrón de

rejilla, proporcionando el balance ideal entre resistencia y ligereza. Para estas piezas se utilizaron soportes en las áreas más complejas para mantener la integridad del diseño durante la impresión.

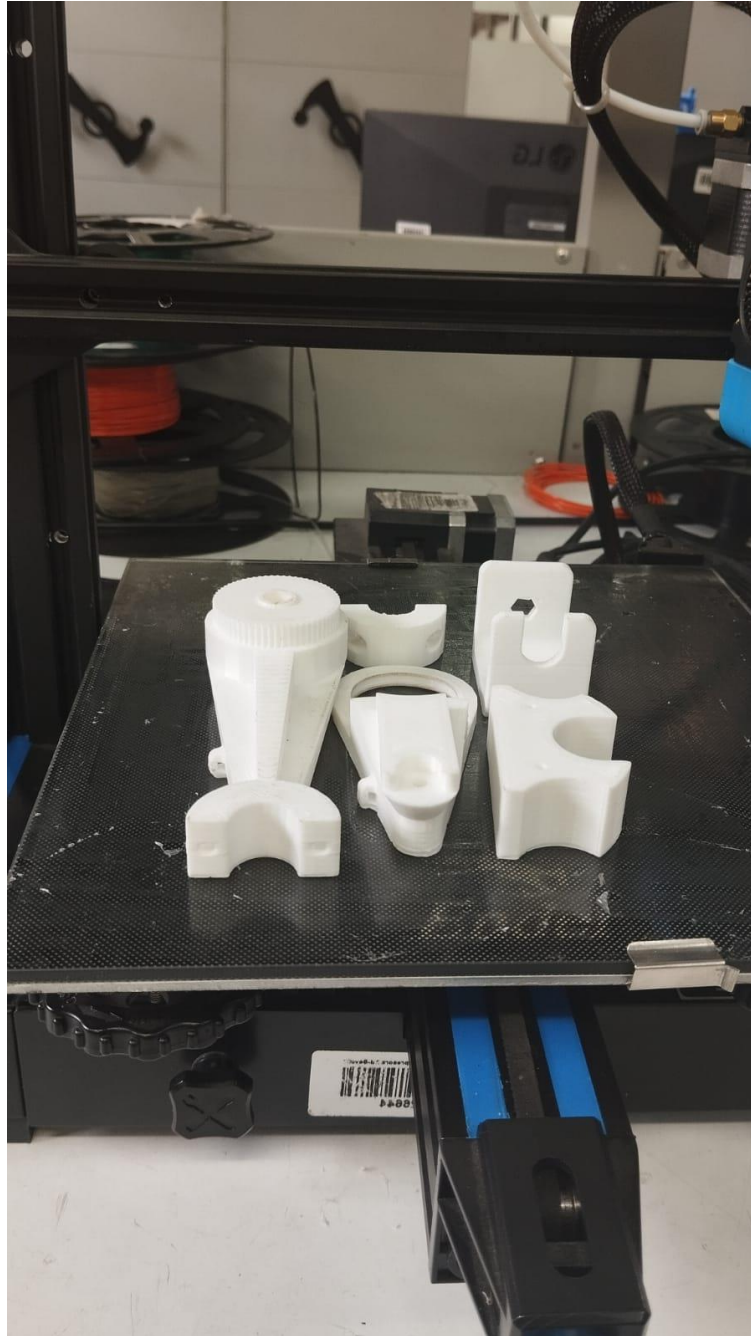


Figura 55. Conjunto de piezas adicionales impresas en PLA
Fuente: diseño propio

Las correas, por otro lado, se imprimieron en TPU, un material flexible adecuado para piezas que requieren elasticidad y capacidad de deformación controlada. La configuración para el TPU incluyó una altura de capa de 0.1 mm y una velocidad reducida de 30 mm/s, con la cama caliente a 50 °C y la boquilla a 220 °C, lo que permitió obtener piezas precisas y resistentes al desgaste, ideales para las partes móviles del ensamblaje.

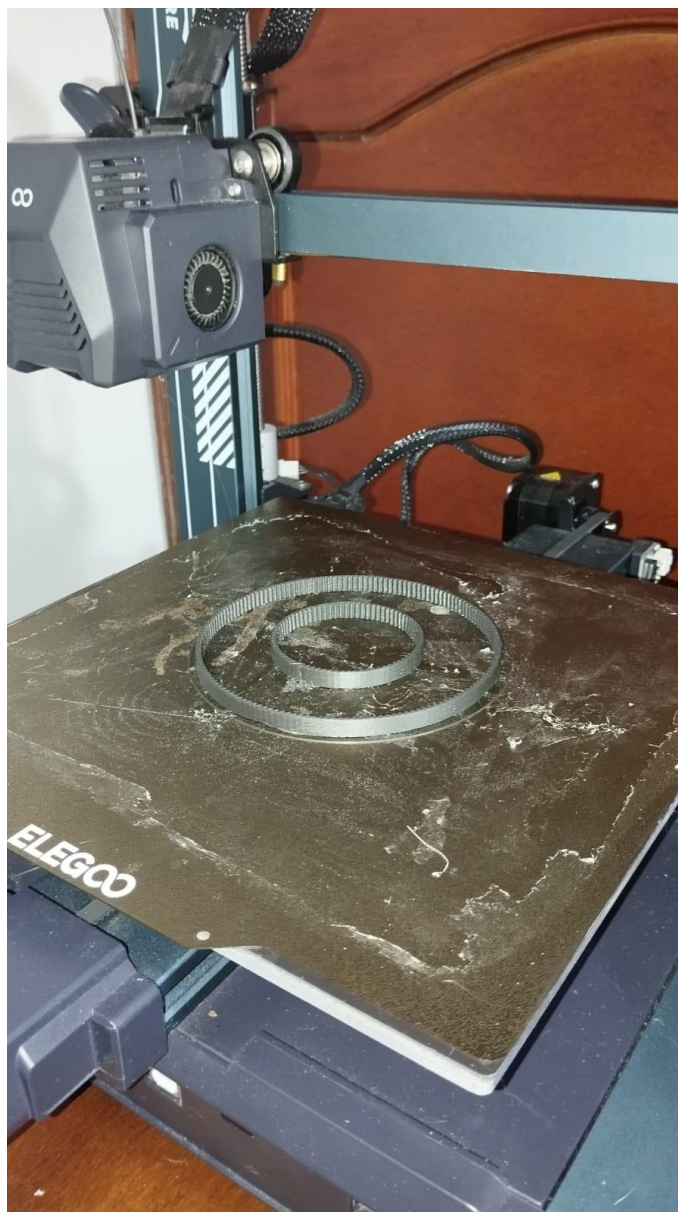


Figura 56. Impresión de correas flexibles en TPU para partes móviles
Fuente: diseño propio

6.2.2. Ensamble físico de piezas. Una vez impresas, se llevó a cabo el ensamble físico de las piezas. Este proceso implicó:

Ensamble del primer eslabón: El proceso inicia con el ensamble del primer eslabón del brazo robótico, que incluye las partes principales que sostendrán el motor y otros componentes. Es esencial garantizar que todos los tornillos estén correctamente insertados para evitar movimientos sueltos.



Figura 57. Ensamble primer eslabón
Fuente: diseño propio

Ensamble del Segundo Eslabón: Se continúa con el ensamble del segundo eslabón, que, aunque más corto, es igualmente importante. Es fundamental asegurar bien los puntos de conexión con el primer eslabón.



Figura 58. Ensamble segundo eslabón
Fuente: diseño propio

Ensamble del Motor del Segundo Eslabón: se procede a colocar el motor en el segundo eslabón del brazo robótico. Es crucial asegurarse de que los tornillos estén bien ajustados y que el motor esté alineado correctamente con las piezas móviles.

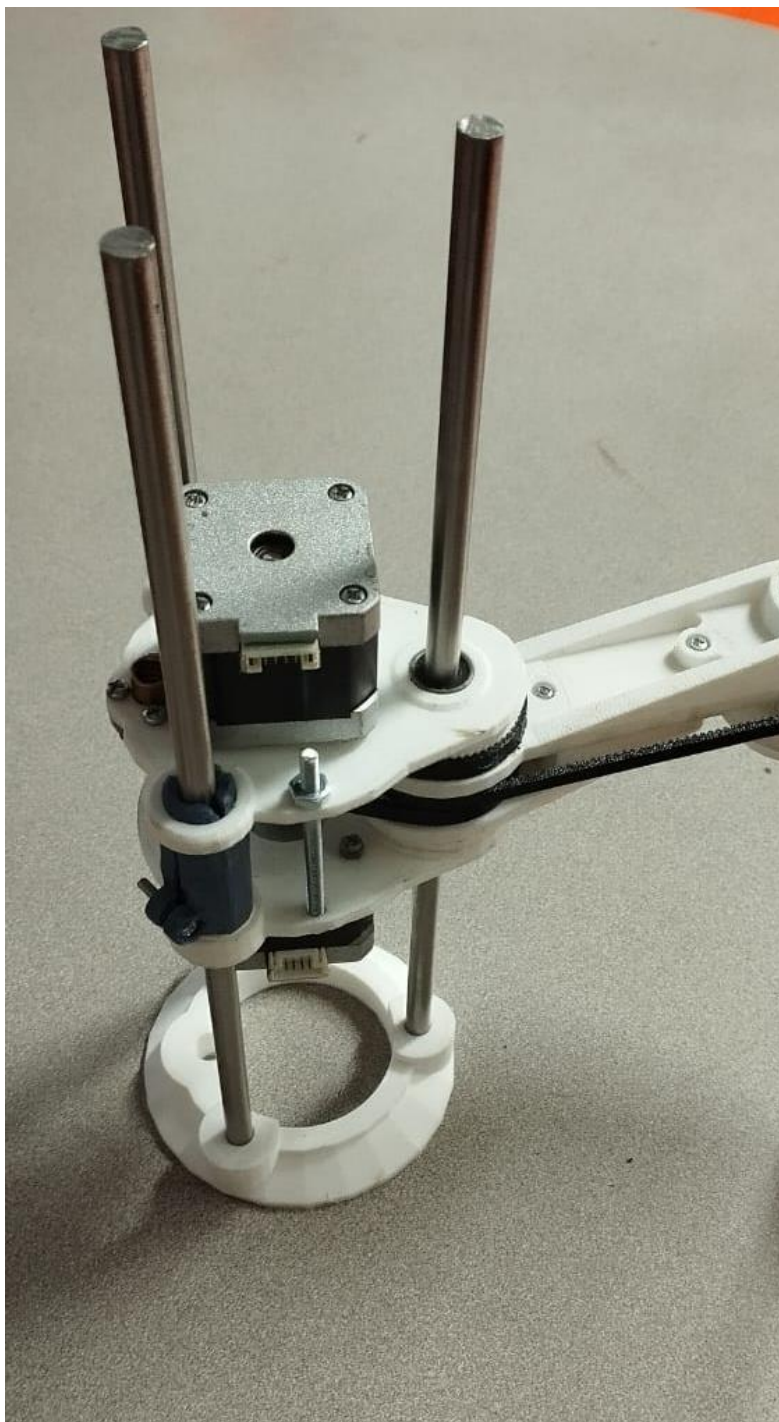


Figura 59. Ensamble motor segundo eslabón
Fuente: diseño propio

Ensamble del Soporte del Eje Z: a continuación, se coloca el soporte del eje Z, que permitirá los movimientos verticales del brazo. Este paso es crucial para el funcionamiento del brazo en los tres ejes, por lo que se deben ajustar todos los componentes del eje con precisión.



Figura 60. Ensamblaje motor eje Z

Fuente: diseño propio

Ajustes Finales: se debe asegurar que todos los componentes estén bien ajustados y alineados. Es recomendable girar y probar manualmente el movimiento del brazo para verificar que no haya fricción ni piezas sueltas.



Figura 61. Ensamblaje final
Fuente: diseño propio

6.2.3. Montaje Electrónico. Esta fase fue fundamental para asegurar el correcto funcionamiento del sistema, y se llevó a cabo siguiendo el diagrama de conexiones establecido.



Figura 62. Conexiones eléctricas
Fuente: diseño propio

Durante el montaje, se realizó la integración de componentes esenciales, tales como motores, drivers, servomotores y la fuente de alimentación. Cada componente fue interconectado de manera precisa para garantizar la operatividad del robot. Se prestó especial atención a la secuenciación de las conexiones, asegurando la coherencia en el funcionamiento de cada parte del sistema robótico.

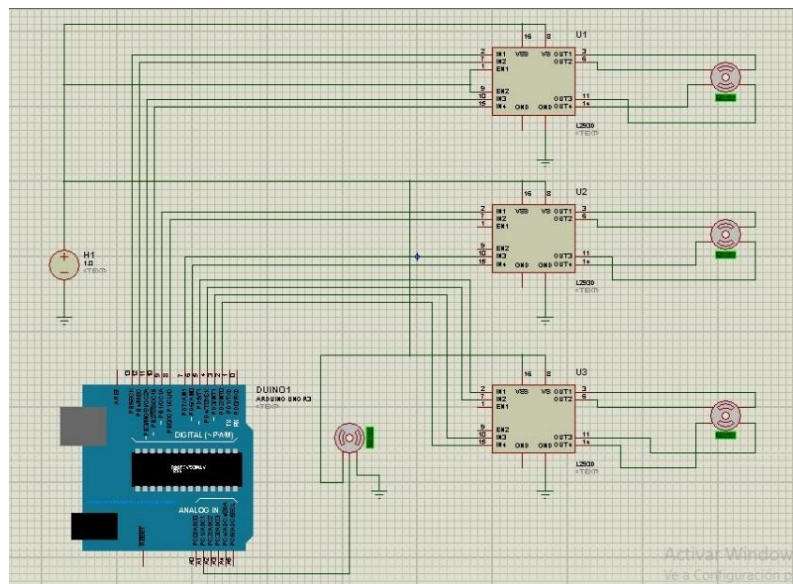


Figura 63. Diagrama eléctrico
Fuente: diseño propio

A continuación, se detallan los resultados del ensamble y la conexión de los diferentes componentes, así como las correcciones realizadas para optimizar su desempeño.

Montaje del CNC SHIELD: Se completó la instalación de los drivers, asegurando que los potenciómetros estuvieran orientados hacia abajo, conforme a las especificaciones del diagrama. Esta disposición garantiza un correcto acceso a los ajustes de calibración.

Conexión de Motores:

Motor 1: controla el movimiento del eslabón número uno. Se realizó la conexión siguiendo el código de colores (negro, verde, azul y rojo), asegurando que la fase correcta estuviera conectada a 2A y 22B.

Motor 2: controla el movimiento del eslabón número dos. Se observó que la conexión inicial no correspondía con el movimiento esperado; por lo tanto, se realizó una inversión en las fases, logrando que el motor girara en la dirección correcta. La secuencia de colores utilizada fue rojo, azul, verde y negro.

Motor 3: este motor, que controla el movimiento del eje Z, fue conectado siguiendo la misma secuencia de colores que el motor 1 (negro, verde, azul y rojo). Esto asegura uniformidad en el sistema de conexiones, facilitando el diagnóstico de problemas futuros.



Figura 64. Montaje eléctrico
Fuente: diseño propio

Conexión del Servomotor: el servomotor fue conectado de acuerdo con las indicaciones. Se aseguró que el pin de señal del servomotor estuviera vinculado adecuadamente al conector Z+, lo que permite un control preciso del movimiento.

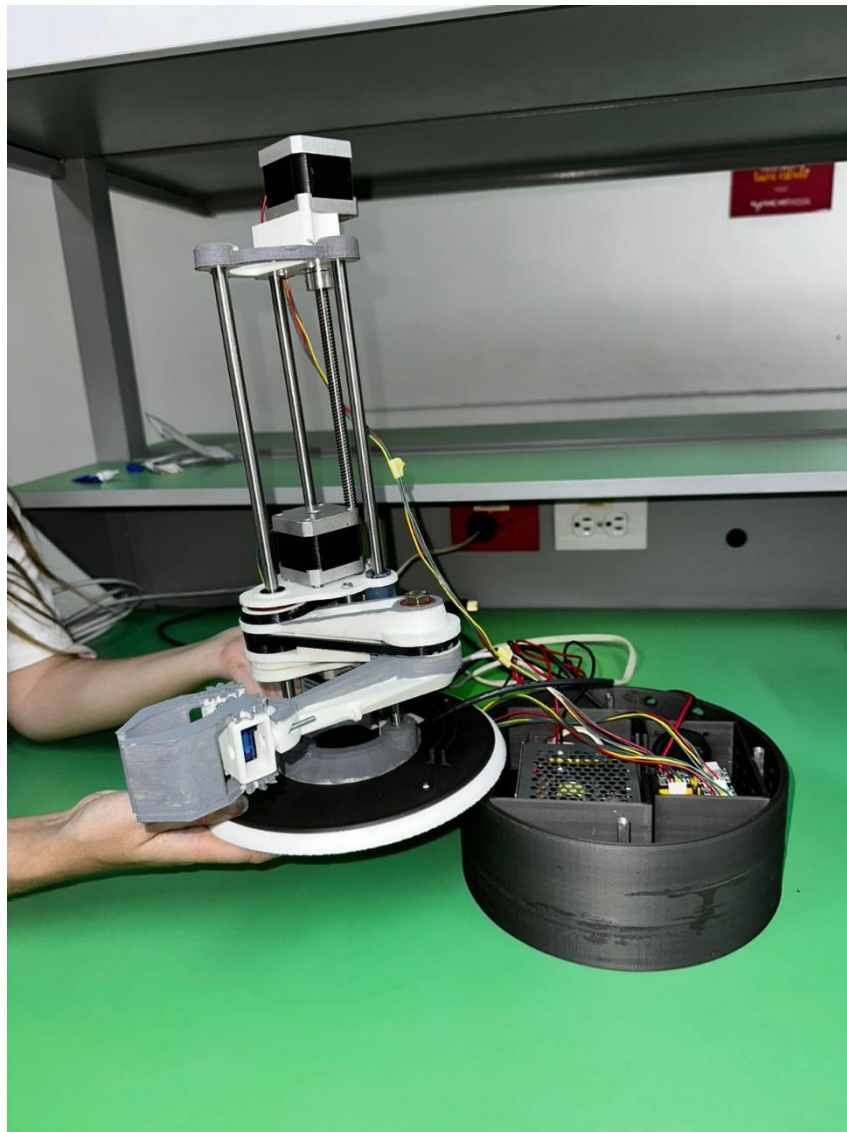


Figura 65. Conexiones eléctricas
Fuente: diseño propio

Alimentación del Sistema: se utilizó una fuente de alimentación de 12 voltios y 3 amperios para energizar el sistema. La conexión se realizó con un acople adecuado, dirigiendo la salida positiva hacia VCC y conectando GND correctamente, lo que garantiza una fuente de energía estable para todos los componentes.

Habilitación del Driver: para activar el driver, se utilizó un jumper para conectar el pin de enable a GND, cumpliendo con el requisito de habilitación y asegurando que todos los drivers estén operativos.

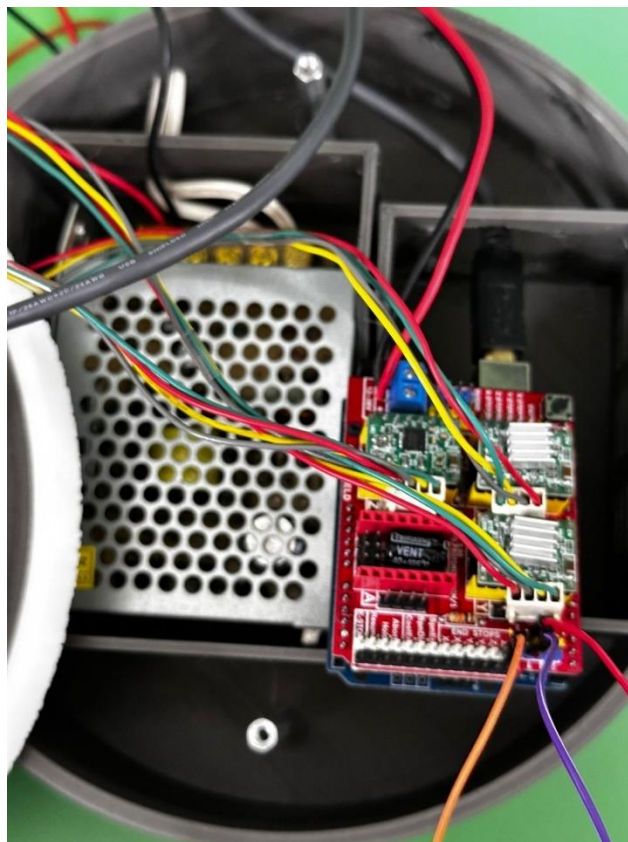


Figura 66. Montaje eléctrico
Fuente: diseño propio

6.3 Implementación de interfaz gráfica para el manejo del módulo “SCARA”

La interfaz gráfica desarrollada para el manejo del brazo robótico SCARA fue creada utilizando la biblioteca Tkinter de Python. Tkinter es una herramienta poderosa que permite la creación de interfaces de usuario (UI) intuitivas y visualmente atractivas. El objetivo principal de esta interfaz es facilitar la interacción del usuario con el brazo robótico, brindando un entorno accesible para el control manual y la programación de movimientos.

6.3.1 Diseño Inicial de la Interfaz. El diseño inicial de la interfaz se centró en ofrecer un control simple pero preciso sobre los movimientos del brazo robótico. La interfaz fue construida con una estructura clara, utilizando controles interactivos y elementos visuales que permiten a los usuarios ajustar los parámetros del robot con facilidad.

Para optimizar la legibilidad, se eligió una paleta de colores oscuros como fondo, con textos y botones en tonos contrastantes de azul futurista. Esto no solo mejora la visualización en entornos con poca luz, sino que también proporciona una sensación moderna y tecnológica, alineada con el propósito del proyecto.

Se incorporó una imagen representativa del brazo robótico, la cual fue ajustada para que fuera fácilmente visible y proporcionara un contexto visual para los controles de la interfaz. Además, se cuidaron aspectos ergonómicos en la disposición de los controles, ubicándolos de manera que fueran fáciles de acceder y usar durante la operación del brazo

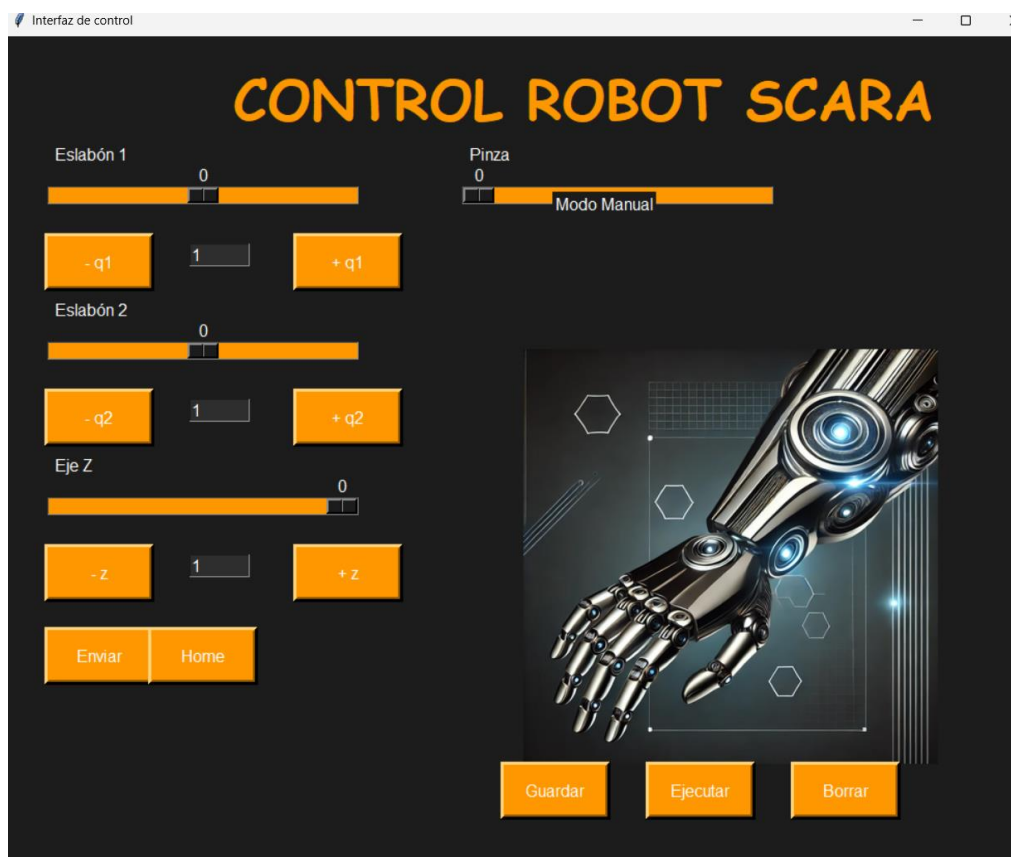


Figura 67. Plataforma Grafica

Fuente: diseño propio

6.3.2 Controles de Movimiento. El elemento central de la interfaz son los controles deslizantes (escalas), que permiten al usuario ajustar los ángulos de los diferentes componentes del brazo robótico: los eslabones (Eslabón 1 y Eslabón 2), el eje Z y la pinza. Estos controles se

diseñaron para ser altamente precisos y fáciles de usar, con un rango de movimiento específico para cada componente:

Tabla 2.
Controles de movimiento

Componente	Rango de Movimiento
Eslabón 1	-90° a 90°
Eslabón 2	-90° a 90°
Eje Z	-130° a 0°
Pinza (gripper)	0° a 100°

Fuente: elaboración propia

Cada uno de estos controles se encuentra acompañado de botones de incremento y decremento, que permiten al usuario ajustar los ángulos de manera manual. Esta opción es útil para realizar ajustes rápidos y precisos sin necesidad de interactuar con los controles deslizantes. Además, cada vez que se ajusta un ángulo, el valor actualizado se refleja de inmediato en la interfaz, lo que garantiza que el usuario tenga un control total sobre la posición del brazo.

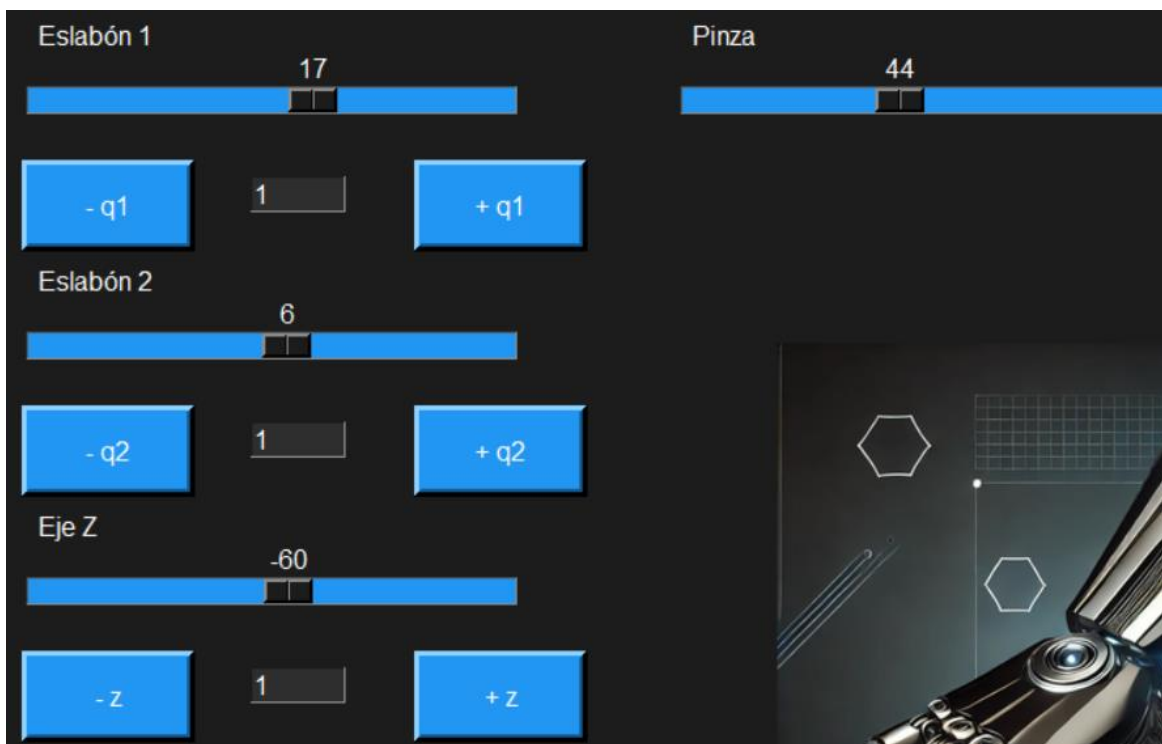


Figura 68. Controles de movimiento
Fuente: diseño propio

6.3.3 Funciones de Control. A lo largo del desarrollo de la interfaz, se integraron diversas funciones clave que mejoran la experiencia del usuario y amplían las capacidades del sistema:

Enviar Posiciones: Un botón que permite al usuario enviar las configuraciones actuales de los ángulos al robot. Esta acción es crucial para aplicar los ajustes realizados de inmediato.

Función Home: Un botón que reposiciona el brazo robótico a su posición inicial. Esto resulta útil cuando se quiere reiniciar el sistema o después de completar una tarea.

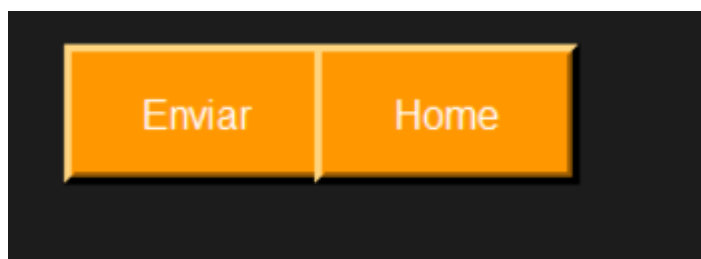


Figura 69. Botones enviar y home

Fuente: diseño propio

Guardar y Ejecutar Secuencias: La función de guardar permite almacenar las posiciones actuales de los ángulos en una lista, lo que facilita la recuperación de configuraciones guardadas para su uso posterior. La función de ejecutar permite al usuario iniciar una secuencia de movimientos previamente almacenados, aumentando la eficiencia en la operación del robot.

Borrar Posiciones: Esta opción permite al usuario borrar las posiciones guardadas y restaurar la interfaz a su estado inicial, proporcionando flexibilidad para nuevos ajustes o secuencias.

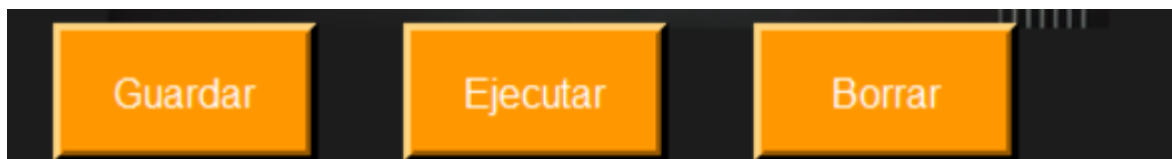


Figura 70. Botones guardar, ejecutar y enviar

Fuente: diseño propio

6.3.4 Comunicación con Arduino. La comunicación entre la interfaz gráfica y el microcontrolador Arduino se establece a través de una conexión serial, lo que permite que los

comandos del usuario sean enviados de manera eficiente al brazo robótico. El código en la interfaz gestiona esta comunicación, enviando las instrucciones de los controles directamente al Arduino. Este microcontrolador, a su vez, se encarga de controlar los motores que mueven el brazo robótico de acuerdo con las configuraciones enviadas.

La interfaz gráfica también incluye un indicador de estado, que muestra al usuario si el sistema está operando en modo manual o ejecutando posiciones preprogramadas. Este feedback visual es esencial para mantener un control adecuado sobre el proceso de operación del brazo, garantizando que el usuario esté siempre informado sobre el estado del sistema.

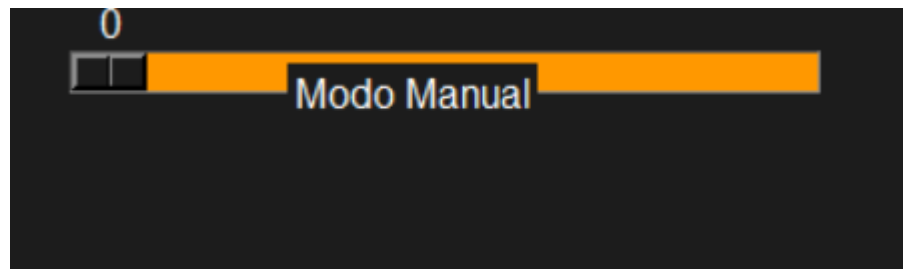


Figura 71. Indicador de estado

Fuente: diseño propio

6.3.5 Últimas Mejoras. Durante el proceso de desarrollo y las pruebas del sistema, se realizaron varias modificaciones para mejorar la usabilidad y funcionalidad de la interfaz gráfica:

Optimización de la disposición de controles: se ajustaron los botones y controles para que fueran más accesibles y estuvieran mejor organizados, lo que facilita la navegación y el uso eficiente del sistema.

Añadido de funcionalidades adicionales: se incluyó la capacidad de ejecutar una secuencia de movimientos guardados repetidamente, lo que permite que el robot realice tareas de manera autónoma sin necesidad de intervención constante.

Mejoras visuales: se refinaron los colores y fuentes para mejorar la estética y la legibilidad, además de asegurarse de que la interfaz fuera funcional en diferentes condiciones de iluminación.

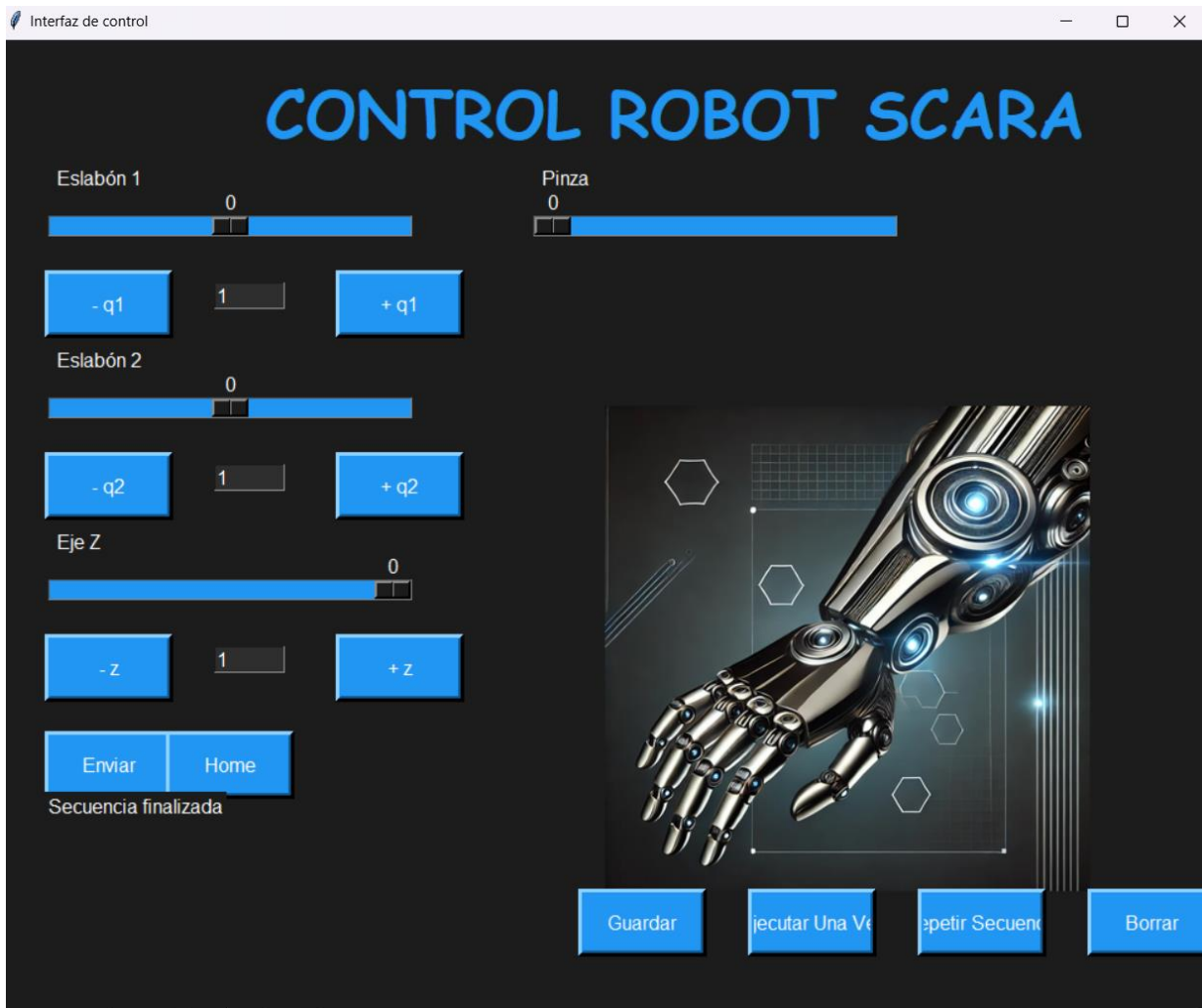


Figura 72. Interfaz Final
Fuente: diseño propio

En el anexo A se puede apreciar un video del funcionamiento de la interfaz gráfica

6.4 Código de control de la interfaz gráfica

6.4.1 Importación de Bibliotecas. Para el desarrollo del control de la interfaz gráfica se usaron diversas librerías, las cuales se detallan a continuación.

- Tkinter: Es la biblioteca estándar de Python para crear interfaces gráficas de usuario (GUI). Proporciona herramientas para crear ventanas, botones, etiquetas, controles deslizantes, etc.
- PIL (Python Imaging Library): Se usa para trabajar con imágenes. En este caso, es necesario para cargar y mostrar la imagen del brazo robótico.
- pyArduino: Es una biblioteca que se utiliza para interactuar con el microcontrolador

```
from tkinter import *
from tkinter import font
from PIL import Image, ImageTk
from pyArduino import *
```

Figura 73. Importación de Bibliotecas

Fuente: diseño propio

6.4.2 Definición de Funciones. Las siguientes funciones se ejecutan cuando el usuario intenta cerrar la ventana de la interfaz, Primero, envía datos al Arduino para asegurarse de que el robot se detenga, luego, cierra la conexión serial con Arduino y finalmente, cierra la ventana de la interfaz gráfica (`root.destroy()`).

```
def onClosing():
    arduino.sendData([0, 0, 0, 0])
    arduino.close()
    print("Arduino desconectado")
    root.destroy()
```

Figura 74. Definición de funciones

Fuente: diseño propio

6.4.3 Funciones de Movimiento. Estas funciones ajustan los ángulos de los eslabones del brazo robótico. Cada función toma el valor actual del ángulo (usando `q1Angle.get()`), lo modifica según el valor ingresado por el usuario en el campo de texto (`q1Value.get()`), y luego actualiza el control deslizante (`q1Scale.set()`).

```
def q2MinusOk():
    q2Angle.set(int(q2Angle.get()) - int(q2Value.get()))
    q2Scale.set(q2Angle.get())
```

Figura 75. Funciones de movimiento
Fuente: diseño propio

6.4.4 Funciones de Control de Secuencias. Guarda la configuración actual de los ángulos en una lista (`secuencia_guardada`) y Muestra un mensaje de estado que indica que la posición se ha guardado correctamente.

Otras funciones usadas fueron:

- `Ejecutar_una_vez()`: Ejecuta la secuencia guardada solo una vez. Si hay secuencias almacenadas, la función comenzará la ejecución desde el inicio y solo la hará una vez.
- `Ejecutar_repetidamente()`: Ejecuta la secuencia guardada en un bucle continuo. La secuencia se repite indefinidamente hasta que se detenga o se presione el botón para interrumpirla.
- `Borrar ()`: Permite borrar todas las secuencias guardadas y restaurar el sistema a su estado inicial, limpiando las configuraciones previas.

```
def guardar():
    secuencia_guardada.append([q1Angle.get(), q2Angle.get(), zAngle.get(), gAngle.get()])
    status.set("Posición guardada en secuencia")
```

Figura 76. Control de secuencias
Fuente: diseño propio

6.4.5. Función de Comunicación con Arduino. Esta función es clave para controlar el flujo de la ejecución de la secuencia de movimientos. Verifica si la secuencia está en ejecución (`isRun`) y si hay más movimientos para ejecutar. Si la secuencia ha terminado y no se debe repetir, detiene la ejecución y muestra el mensaje "Secuencia finalizada". Si la secuencia debe repetirse (`repetir = True`), reinicia el índice `k` y continúa la ejecución desde el principio.

```

def callback():
    global isRun, k
    if isRun and k < len(secuencia_guardada):
        arduino.sendData(secuencia_guardada[k])
        k += 1
    else:
        if repetir:
            k = 0 # Reinicia el índice para repetir la secuencia
        else:
            arduino.sendData([q1Angle.get(), q2Angle.get(), zAngle.get(), gAngle.get()])
            k = 0
            isRun = False
            status.set("Secuencia finalizada")
            root.after(1000, callback)

##### COMUNICACION SERIAL #####
port = "COM7"
arduino = serialArduino(port)

```

Figura 77. Comunicación con Arduino

Fuente: diseño propio

6.4.6 Configuración de la Interfaz Gráfica. Se crea la ventana principal con un tamaño de 1000x800 píxeles y un título en la parte superior, lo que sirve como contenedor para todos los elementos gráficos. Se carga y redimensiona la imagen del brazo robótico a 400x400 píxeles. Esta imagen se coloca en la interfaz en una ubicación específica utilizando un widget Label para mostrar visualmente el robot. Se añaden controles deslizantes (escala) para ajustar los ángulos de los eslabones del brazo robótico, como el Eslabón 1, con un rango de -90 a 90 grados. Estos controles permiten manipular el robot con precisión. Se define una función para crear botones personalizados con acciones como ejecutar movimientos o guardar configuraciones. Además, los botones tienen un efecto visual que cambia de color cuando el cursor pasa sobre ellos, mejorando la interacción del usuario.

```

root = Tk()
root.geometry("1000x800")
root.title("Interfaz de control")

# Configuración de colores y fuente
bg_color = "#1C1C1C"
fg_color = "#2196F3" # Azul futurista
font_style = ("Helvetica", 12)
font_style_title = ("Comic Sans MS", 40, "bold")

# Configuración de la interfaz
root.configure(bg=bg_color)

# Imagen del brazo robótico
image_path = "brazo.png"
robotic_arm_image = Image.open(image_path)
robotic_arm_image = robotic_arm_image.resize((400, 400), Image.LANCZOS)
robotic_arm_photo = ImageTk.PhotoImage(robotic_arm_image)

image_label = Label(root, image=robotic_arm_photo, bg=bg_color)
image_label.place(x=500, y=300)

# Título
title_frame = Frame(root, bg=bg_color)
title_frame.place(x=200, y=10)
title = Label(title_frame, text="CONTROL ROBOT SCARA", font=font_style_title, bg=bg_color, fg=fg_color, padx=20, pady=10)
title.pack()

##### Eslabón 1 #####
qlAngle = IntVar()
qlScale = Scale(root, command=getAngle, label="Eslabón 1", font=font_style, from_=-90, to=90, orient=HORIZONTAL, length=300,
                bg=bg_color, fg="white", troughcolor=fg_color, highlightbackground=bg_color)
qlScale.place(x=40, y=100)

def create_custom_button(text, command, x, y):
    button = Button(root, text=text, command=command, height=2, width=10,
                   bg=fg_color, fg="white", font=font_style, borderwidth=0)
    button.place(x=x, y=y)
    button.bind("<Enter>", lambda e: button.config(bg="#64B5F6"))
    button.bind("<Leave>", lambda e: button.config(bg=fg_color))
    button.config(relief=RAISED, bd=5)
    return button

```

Figura 78. Configuración interfaz
Fuente: diseño propio

6.4.7 Finalización y Actualización de la Interfaz. El código finaliza con la ejecución del ciclo principal de la interfaz gráfica:

```

root.protocol("WM_DELETE_WINDOW", onCloseing)
root.after(100, callback)
root.mainloop()

```

Figura 79. Finalización
Fuente: diseño propio

En el anexo B se presenta el código completo de la interfaz gráfica.

6.5 Código de control del brazo robótico SCARA

El código está diseñado para controlar un brazo robótico compuesto por tres motores paso a paso (para los eslabones y el eje Z) y un servo motor para la pinza. La comunicación con el

sistema se realiza a través del puerto serial, donde se reciben datos que determinan las posiciones de los motores y el ángulo de la pinza.

Se utilizan tres instancias de la clase `AccelStepper` para controlar los motores paso a paso de los eslabones y del eje Z. Cada motor tiene pines de paso y dirección asignados, lo que permite controlarlos con precisión. Para cada motor, se configura la velocidad máxima y la aceleración a través de las funciones `setMaxSpeed()` y `setAcceleration()`. También se inicializa la posición de los motores con `setCurrentPosition(0)` para asegurarse de que todos los motores comiencen en la posición cero.

Los ángulos de los eslabones y el eje Z se convierten a pasos utilizando las fórmulas de conversión basadas en la relación entre los dientes de las poleas y el número de pasos por vuelta del motor. Esto asegura que el movimiento de los motores se corresponda con el movimiento esperado de los eslabones y el eje Z. La variable `q1AngleToSteps` y otras similares definen cómo se deben transformar los valores de los ángulos a pasos de motor.

Cuando el código recibe nuevos datos a través de la comunicación serial, estos datos se procesan y se utilizan para actualizar las posiciones de los motores. Se utiliza la función `moveTo()` para indicar a los motores la nueva posición a la que deben dirigirse. Los motores se mueven de manera sincronizada hasta alcanzar la posición deseada, utilizando la función `run()` dentro de un bucle `while` que asegura que todos los motores se detengan cuando lleguen a su posición.

El servo motor de la pinza se controla a través de la función `myServo.write()`, que ajusta el ángulo del servo según el valor recibido en los datos. El ángulo está mapeado de 0 a 100 en el valor de entrada, que luego se convierte a un rango de 0 a 80 para controlar el movimiento de la pinza.

La función `serialEvent()` se ejecuta cada vez que hay datos disponibles en el puerto serial. Se lee cada carácter recibido y se almacena en la variable `inputString`. Cuando se recibe una línea completa (indicada por el carácter de salto de línea `\n`), la variable `stringComplete` se establece

en true, lo que permite al código procesar los datos y convertirlos en posiciones para los motores.

Los datos recibidos a través del puerto serial son separados por comas, y cada valor corresponde a una posición para un motor o el ángulo de la pinza. Estos valores se almacenan en el arreglo datos[]. Los valores son convertidos de texto a flotantes y luego se asignan a las variables correspondientes que controlan las posiciones de los motores y la pinza.

El código permite mover los tres motores paso a paso y el servo motor de la pinza de manera precisa y controlada según los valores recibidos a través de la comunicación serial. Cada motor se mueve hasta la posición deseada, y la pinza se ajusta para abrirse o cerrarse según se necesite.

En el anexo C se presenta el código completo desarrollado.

6.6 Pruebas exhaustivas al prototipo de brazo robótico SCARA

Previo a las pruebas realizadas con estudiantes, se llevaron a cabo pruebas internas para evaluar el ensamblaje y la funcionalidad de la interfaz gráfica del brazo robótico SCARA. Durante estas pruebas, se identificaron varios aspectos a mejorar tanto en la estructura física del brazo como en la interfaz de usuario. En cuanto a la interfaz gráfica, se observó la necesidad de añadir nuevos botones y funciones que podrían optimizar la interacción y el control del robot, mejorando la experiencia de uso.

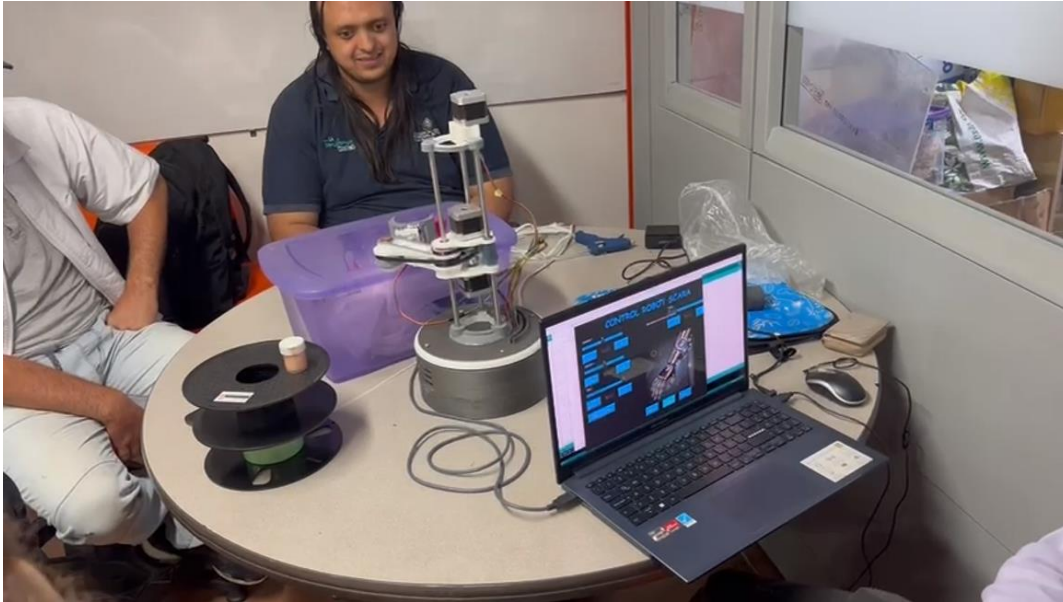


Figura 80. Finalización
Fuente: diseño propio

A nivel estructural, se detectó que las pinzas del brazo presentaban inconvenientes al unirse, ya que no engranaban adecuadamente, lo cual afectaba la precisión de las operaciones. Además, se identificó la necesidad de analizar el sonido emitido por los motores y se consideró que el código podría ser ajustado para suavizar este sonido, explorando la posibilidad de implementar el método de micropasos de 1/16 en el código de Arduino para aumentar la precisión en los movimientos.

En el anexo D se presentan videos de pruebas del brazo funcional.

También se evidenció la necesidad de proporcionar mayor estabilidad al brazo mediante una base más sólida. A partir de esta observación, se diseñó una carcasa que incluye una pieza de hierro en su interior, la cual aporta estabilidad y brinda al brazo una apariencia más robusta. Con estos aspectos observados, el prototipo quedó listo para ser evaluado en las pruebas con estudiantes, permitiendo analizar su ensamble y funcionamiento en un entorno educativo.

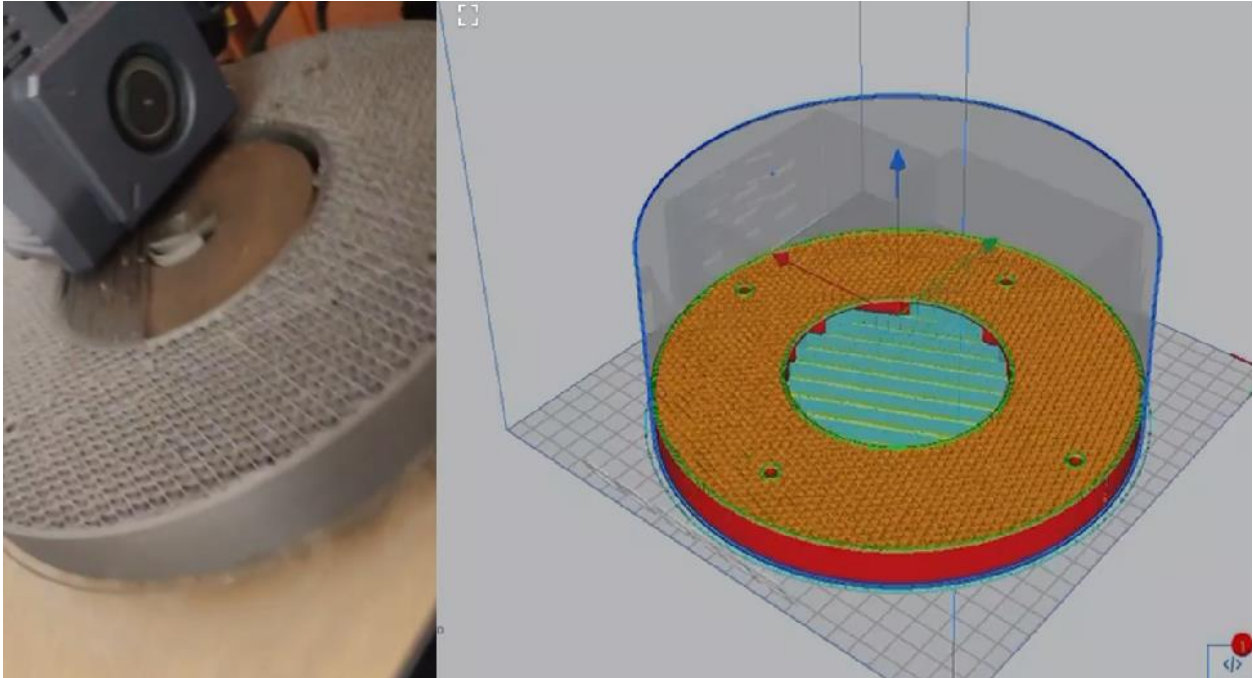


Figura 81. Base
Fuente: diseño propio

Posteriormente se llevaron a cabo pruebas exhaustivas al prototipo del brazo robótico SCARA con estudiantes de diferentes grados y técnicas del Técnico Industrial Pascual Bravo, con el fin de evaluar la facilidad y optimización del proceso de ensamble. Los estudiantes fueron seleccionados de diversas áreas técnicas, lo que permitió observar la percepción de un grupo heterogéneo en cuanto a la complejidad del montaje.



Figura 82. Pruebas
Fuente: diseño propio

Durante las pruebas, se les mostró el proceso de ensamblaje a través del software Fusión 360, lo que permitió que los estudiantes pudieran visualizar claramente las instrucciones paso a paso y comprender mejor las interacciones entre las piezas. Esta herramienta visual fue fundamental para asegurar una correcta ejecución del ensamble, especialmente en los estudiantes con menos experiencia en robótica. El ensamblaje fue realizado por los estudiantes, quienes se mostraron entusiasmados y motivados durante el proceso. Los comentarios fueron muy positivos, ya que la mayoría de los participantes encontraron el ensamblaje relativamente fácil y accesible. Esto indicó que el diseño del prototipo permite que personas con distintos niveles de conocimiento técnico puedan montarlo sin mayores dificultades.

Una vez finalizadas las pruebas, se les entregó un formato con preguntas específicas para que pudieran expresar su opinión sobre el proceso. El formato fue diseñado para recoger información detallada sobre la facilidad del ensamblaje, las dificultades encontradas y las sugerencias de mejora. Esta retroalimentación fue valiosa para identificar aspectos del prototipo

que podrían optimizarse, así como para confirmar que el diseño actual es funcional y adecuado para su propósito educativo.

Johan Adrian, Mecanica Automatiz

Cuestionario Pruebas

1. ¿Qué partes del robot encontraste más difíciles de ensamblar y por qué?

la pieza de la parte mas difícil.
ya que el ensamblaje tambien
dependia del eje del servomotor
y una buena tuerca de atornillar

2. ¿Sentiste que había suficiente claridad en las instrucciones de montaje? ¿Qué mejorarías en ellas para facilitar el ensamblaje?

hubo una buena claridad, el paso a
paso se entendio mejorata el en-
samblaje de algunos tornillos, como
los de la pieza

3. ¿Observaste alguna pieza o diseño que podría modificarse para que el ensamblaje sea más rápido y sencillo?

Ninguna

Figura 83. Formato de Preguntas

Fuente: diseño propio

Los resultados obtenidos, los cuales están disponible en el Anexo E, confirmaron que el prototipo de brazo robótico SCARA está bien diseñado en términos de facilidad de ensamble, lo que lo convierte en una herramienta adecuada para entornos educativos. A partir de las pruebas, se identificaron áreas de mejora que permitirán hacer el proceso de ensamble aún más eficiente y accesible para los futuros usuarios.



Figura 84. Fotos de las pruebas realizadas con estudiantes

Fuente: diseño propio

Otras fotografías del proceso pueden verse en el Anexo F.

6.7 Ajustes Finales

Se han realizado cambios significativos para mejorar tanto la estabilidad estructural como la funcionalidad del brazo robótico SCARA. Se diseñó una base sólida que proporciona mayor firmeza al brazo durante su operación, reduciendo las vibraciones y aumentando la estabilidad en movimientos complejos.



Figura 85. Ajustes Finales Brazo Scara
Fuente: diseño propio

Para completar el sistema de soporte, se continúa con la construcción de una carcasa superior que será ensamblada junto con esta base, creando un conjunto robusto que brinda estabilidad y aporta una apariencia profesional e imponente al prototipo.



Figura 86. Diseño de base y carcasa
Fuente: diseño propio

En el anexo G se pueden visualizar fotos y videos del prototipo final y del diseño e impresión de la base y carcasa.

En cuanto a la interfaz gráfica, se han añadido mejoras importantes orientadas a optimizar la experiencia del usuario y garantizar una operación segura y precisa. Entre estas mejoras destaca la incorporación de un botón de parada de emergencia, el cual permite detener el brazo de manera inmediata en caso de alguna eventualidad, aumentando así la seguridad del entorno y del propio sistema. También se añadió la funcionalidad de visualizar la secuencia de movimientos guardada, lo cual permite al usuario observar y controlar en tiempo real la trayectoria programada del brazo, facilitando la validación y ajuste de los movimientos antes de su ejecución. Estas adiciones aportan un mayor control sobre el robot y mejoran la supervisión del proceso de trabajo.

Además, con el objetivo de hacer más accesible el ensamblaje del brazo robótico, se evalúa la posibilidad de desarrollar un manual interactivo utilizando Blender. Este manual permitiría a los usuarios visualizar el proceso de ensamblaje paso a paso en un entorno 3D, proporcionando una

guía visual intuitiva y amigable para el usuario. Un manual interactivo ayudaría a que el montaje sea más comprensible, especialmente para aquellos que tienen menos experiencia en robótica, ya que ofrecería instrucciones detalladas con imágenes y animaciones que facilitarían el ensamblaje de cada componente.

A futuro, el desarrollo del proyecto continuará con mejoras en el proceso de ensamblaje, en la estructura del brazo y en el código de control. En el ensamble, se explorará la posibilidad de simplificar y optimizar el montaje de componentes, de manera que sea más intuitivo y accesible para los usuarios. Estructuralmente, se revisarán las pinzas y articulaciones para asegurar un mejor engranaje y precisión en cada movimiento, además de estudiar el uso de materiales y mecanismos que aporten mayor durabilidad.

En el ámbito de software, se proyecta optimizar el código de control, incorporando técnicas como el uso de micropasos (1/16) para lograr una mayor precisión en los movimientos y reducir el ruido de los motores, mejorando la experiencia auditiva durante la operación. La interfaz gráfica también seguirá siendo mejorada, añadiendo funciones que aumenten su funcionalidad y accesibilidad, haciendo que el sistema sea cada vez más eficiente y fácil de manejar.

Estos ajustes y desarrollos adicionales reflejan el compromiso con la evolución continua del proyecto, encaminándolo a convertirse en un prototipo de referencia para la robótica educativa y la investigación. Con cada iteración, el sistema se optimizará en términos de precisión, estabilidad y facilidad de uso, consolidándose como una herramienta de aprendizaje avanzada y accesible.

7. Conclusiones

Se desarrolló un kit robótico tipo "SCARA" con interfaz gráfica asequible, educativo y fácil de usar, mediante técnicas de prototipado rápido de productos, implementación electrónica y robotización, destinado a la enseñanza y formación en los campos de la robótica y la automatización, de investigadores y estudiantes de las regiones donde la Institución Universitaria Pascual Bravo hace presencia.

Lo anterior fue posible a partir del diseño del prototipo de brazo robótico "SCARA" mediante herramientas CAD, optimizando tanto la eficiencia, como los costos que garantizan su accesibilidad; del mismo modo se desarrolló el prototipo de brazo robótico utilizando materiales asequibles y técnicas de fabricación eficientes, para su implementación en contextos educativos; fue posible implementar una interfaz gráfica para el manejo del módulo "SCARA" mediante Python, esto permite una operación intuitiva y eficiente del brazo robótico en entornos educativos. Finalmente se realizaron pruebas exhaustivas al prototipo por medio de la evaluación de su precisión, velocidad y estabilidad, que permitió la identificación de posibles mejoras y optimizaciones.

El desarrollo de un kit robótico tipo SCARA asequible y educativo ha demostrado ser una herramienta valiosa para acercar la robótica a estudiantes en regiones con recursos limitados. Este proyecto ha contribuido a reducir la brecha tecnológica al facilitar el acceso a una tecnología que usualmente está restringida a instituciones con mayores recursos. Gracias a la implementación de este kit, los estudiantes pueden tener una experiencia práctica en el campo de la robótica, lo cual fomenta su interés y habilidades en esta área clave para el futuro.

La utilización de herramientas de diseño CAD, combinada con técnicas de impresión 3D y componentes electrónicos accesibles, ha resultado en un prototipo funcional y replicable. Los materiales elegidos, como el PLA y el TPU, así como la programación con Arduino, permitieron desarrollar un kit de bajo costo y fácil de ensamblar, sin comprometer la precisión y funcionalidad del brazo robótico. Esta metodología también ha mostrado que el diseño puede ser

replicado en diferentes entornos, promoviendo su potencial para ser implementado en otras instituciones educativas interesadas en fortalecer la enseñanza en robótica.

Al interactuar directamente con el kit SCARA, los estudiantes no solo adquieren conocimientos teóricos, sino que también desarrollan habilidades prácticas en áreas como la cinemática, la programación y el control de sistemas automatizados. Esta experiencia práctica ha sido fundamental para que los estudiantes comprendan mejor los conceptos complejos de la robótica, facilitando el desarrollo de competencias críticas como la resolución de problemas y la innovación. El enfoque experiencial ha demostrado ser eficaz en la enseñanza de conceptos de STEM, haciendo que el aprendizaje sea más significativo y duradero.

Este proyecto no solo tiene un impacto educativo, sino que también contribuye al desarrollo social y tecnológico de las comunidades rurales donde se implementa. Al fomentar el acceso a la educación tecnológica, el proyecto empodera a las comunidades y abre oportunidades para que los jóvenes exploren caminos académicos y profesionales en áreas técnicas. De esta forma, el kit SCARA se convierte en una herramienta que no solo educa, sino que también motiva a los estudiantes a aspirar a una formación en áreas que pueden impactar positivamente sus comunidades.

8. Recomendaciones

A medida que el kit SCARA se integre en entornos educativos reales, se recomienda realizar ajustes adicionales en el diseño, especialmente en las áreas de transmisión y precisión de los movimientos, para optimizar su rendimiento en tareas específicas de manipulación y ensamblaje. Para mejorar la precisión en los movimientos, se sugiere ajustar la programación de los motores utilizando micropasos, como el método de 1/16, para obtener una mayor precisión y suavidad en la respuesta del brazo robótico. Esto facilitará su uso en actividades que requieran exactitud y minuciosidad.

Las pruebas han demostrado que las pinzas actuales presentan inconvenientes en su ensamble, ya que no engranan adecuadamente. Se recomienda un rediseño de las pinzas o el uso de componentes de mayor precisión que aseguren un acoplamiento efectivo y sin errores. Este ajuste optimizará las tareas de agarre y manipulación, mejorando la eficiencia y evitando desajustes que puedan afectar el rendimiento del brazo.

Para garantizar la estabilidad del kit en funcionamiento, es esencial que el diseño de la base sea lo suficientemente sólido como para soportar los movimientos del brazo sin vibraciones indeseadas. En este sentido, se sugiere una base robusta y una carcasa con materiales pesados, como el hierro, que brinden firmeza y mayor estabilidad. Esta estructura no solo mejorará la precisión de las operaciones, sino que también ofrecerá una plataforma segura y confiable en el contexto educativo.

Se considera relevante desarrollar un manual de ensamble interactivo en 3D, utilizando herramientas como Blender, para guiar de manera visual el montaje paso a paso del brazo SCARA. Este recurso facilitaría a estudiantes y docentes la comprensión de cada parte y mecanismo, haciéndolo un proceso más intuitivo y fomentando el aprendizaje activo en robótica. La posibilidad de interactuar con un modelo 3D permitiría a los usuarios adquirir mayor conocimiento sobre la estructura interna y el funcionamiento del sistema.

En cuanto a la interfaz gráfica, se sugiere continuar con la implementación de mejoras, incorporando funciones adicionales que permitan ajustar la velocidad de los movimientos y modificar la secuencia de trabajo en tiempo real. La inclusión de un botón de parada de emergencia y de la opción de visualizar la secuencia guardada ya han sido avances importantes, pero se recomienda seguir refinándolos para ofrecer una interacción segura y optimizada en cada situación de uso.

Para asegurar el máximo impacto educativo, se considera fundamental que los instructores reciban capacitación en el uso y programación del kit SCARA. Este entrenamiento permitirá que los docentes aprovechen todas las funciones del sistema y que los estudiantes logren aplicar conceptos de robótica y automatización de manera práctica y efectiva. La capacitación debe abarcar temas de programación, mantenimiento básico y resolución de problemas comunes, asegurando un uso eficiente y seguro del kit en las aulas.

Dado que el ensamble del kit SCARA debe ser accesible para estudiantes de diferentes niveles de conocimiento, se recomienda realizar una revisión detallada del proceso de ensamble, buscando simplificar pasos y optimizar las instrucciones. Esto facilitará que los usuarios puedan montar el equipo de manera autónoma, promoviendo la independencia y el aprendizaje práctico desde el inicio de la experiencia.

Las pruebas realizadas han evidenciado que el sonido de los motores puede ser mejorado para ofrecer un entorno más silencioso y cómodo. Se sugiere explorar ajustes en el código que permitan reducir el ruido durante el funcionamiento, lo cual beneficiará la concentración de los estudiantes al trabajar con el equipo en un ambiente de aula.

Para aquellos estudiantes que se inician en el ámbito de la robótica, sería beneficioso añadir una opción de programación visual en la interfaz gráfica del kit SCARA. Esto permitiría a los principiantes crear secuencias de movimiento de manera intuitiva y comprender conceptos básicos de robótica a través de una herramienta accesible y amigable.

Se sugiere implementar un sistema de retroalimentación continua en el que tanto instructores como estudiantes puedan registrar sus experiencias y sugerencias de mejora sobre el uso del brazo SCARA en el aula. Esto permitirá realizar ajustes periódicos en el diseño, la programación y el uso del equipo, asegurando que el kit evolucione y se adapte a las necesidades educativas en constante cambio. Además, se recomienda crear una guía completa de funcionamiento y mantenimiento, acompañada de recursos visuales y ejemplos prácticos, para facilitar la familiarización de los usuarios con el equipo y maximizar su efectividad como herramienta de aprendizaje en robótica y automatización.

9. Referencias bibliográficas

- Aguirre Cerrillo, F. A., Orozco Magdaleno, E. C., Lara González, J. A., & Luna Castro, A. (2016). Prototipo de robot Scara con base móvil. *Multidisciplinas de la Ingeniería-Universidad Autónoma de Nuevo León*. Obtenido de <https://mdi.uanl.mx/index.php/revista/article/view/122/102>
- Applied-Motion*. (11 de 10 de 2023). Obtenido de ¿Qué significan los tamaños NEMA?: <https://www.applied-motion.com/s/article/what-do-nema-sizes-mean>
- Caparroso, I. O. (1999). *Una introducción a la robótica industrial*. Ciencia E Ingeniería Neogranadina. Obtenido de <https://doi.org/10.18359/rcin.1410>
- Castillo Castañeda, E., & Rodríguez Vázquez, A. (2011). *LOCALIZACIÓN DEL ELEMENTO EFECTOR DEL ROBOT PARALELO DE TRES GRADOS DE LIBERTAD PARALIX*. Instituto Politécnico Nacional. Obtenido de <http://www.repositoriodigital.ipn.mx/handle/123456789/12588>
- Cortés, F. R. (2020). *Robótica: Control de Robots Manipuladores*. Marcombo.
- EDS Robotics . (04 de 23 de 2021). *EDS Robotics*. Obtenido de ¿Qué es la robótica?: <https://www.edsrobotics.com/blog/que-es-la-robotica/>
- EDUCASE. (2021). 2021 EDUCAUSE Horizon Report. *Librería EDUCAUSE*. Obtenido de <https://library.educause.edu/-/media/files/library/2021/4/2021hrteachinglearning.pdf?la=en>
- Ferrada, C., Carrillo-Rosúa, F. J., Díaz-Levicoy, D., & Silva-Díaz, F. (2020). La robótica desde las áreas STEM en Educación Primaria: una revisión sistemática. *Education in the Knowledge Society (EKS)*, 21. Obtenido de <https://revistas.usal.es/tres/index.php/eks/article/view/eks20202122/22644>
- Guedes, R. J. (30 de 05 de 2024). *Blog de Tecnologías*. Obtenido de Blog de Tecnologías: <https://www3.gobiernodecanarias.org/medusa/ecoblog/rsuagued/arduino/>
- Jara, C. A. (9 de 2008). Jara, Carlos A . *Análisis del espacio de trabajo de un robot paralelo de 3RRR*. Obtenido de <http://hdl.handle.net/10045/10156>
- Jaramillo, D. M. (30 de 05 de 2024). *programarfacil*. Obtenido de programarfacil: https://programarfacil.com/blog/arduino-blog/a4988-drv8825-tmc2209/#Chip_del_driver_A4988

- Jimenez Robayo , C. E. (2009). DISEÑO Y CONSTRUCCIÓN DE UN ROBOT TIPO SCARA CON SISTEMA DE VISION DE MÁQUINA, QUE PUEDA SER UTILIZADO COMO HERRAMIENTA DIDÁCTICA EN ASIGNATURAS DE ROBÓTICA Y CONTROL. *Repository Unimilitar*.
- Mineducación. (2004). *Ministerio de Educación Nacional*. Obtenido de Nuevas tecnologías al servicio de la educación: <https://www.mineducacion.gov.co/1621/article-87399.html>
- Rodríguez, E. (2009). *Dialnet*. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=3086514>
- Romero Ariza, M. (2014). El aprendizaje experiencial y las nuevas demandas formativas. *Antropología Experimental l Universidad de Jaén*. Obtenido de <https://revistaselectronicas.ujaen.es/index.php/rae/article/view/1970>
- Soriano , A., Marín, L., Casalilla, J., Valera, A., Valles, M., & Albertos, P. (2013). PlaPlataforma robótica de bajo coste y recursos limitados basada en Arduino y dispositivos moviles. Ponencia presentada en las XXXIV Jornadas de Automática. *Terrassa*. Obtenido de http://wks.gii.upv.es/cobami/files/MV_JJAAPlataforma%20Arduino.pdf
- Vizcarra Sánchez, Y. A., & Vizcarra Gavilán, A. M. (2021). El laboratorio portátil: herramienta efectiva de enseñanza de la química en entornos rurales. *Universidad Nacional Autónoma*. Obtenido de <https://www.revistas.unam.mx/index.php/req/article/view/72724/69885>

10. Bibliografía

- J y Patiño Cifuentes C, M. C. J. M. N. (2019). Desarrollo de un brazo robótico Scara clasificador de objetos (Institución Universitaria Antonio José Camacho). Institución Universitaria Antonio José Camacho. Recuperado de <https://repositorio.uniajc.edu.co/entities/publication/be5dcec0-632e-460a-80ca-82cbf0e7512e>
- Pareja Aparicio, R. (2017). Estudio para la puesta en marcha de un robot SCARA AdeptThree-XL (UPV). UPV. Recuperado de <https://riunet.upv.es/bitstream/handle/10251/112648/PAREJA%20-%20Estudio%20para%20la%20puesta%20en%20marcha%20de%20un%20robot%20SCARA%20AdeptThree-XL.pdf>
- Pineda, J. C., y Yopez, D. (2018). DISEÑO, MODELAMIENTO E IMPLEMENTACIÓN DE UN ROBOT TIPO SCARA DE TRES GRADOS DE LIBERTAD (Universidad Tecnológica de Pereira). Universidad Tecnológica de Pereira. Recuperado de <https://repositorio.utp.edu.co/server/api/core/bitstreams/c4284d19-d099-46bf-b338-9c593de88000/content>
- Proyectos, J. C. (s/f). Cinemática Directa Robot SCARA por Algoritmo de Denavit Hartenberg y Matlab. Recuperado el 11 de noviembre de 2024, de https://www.youtube.com/watch?v=jo_aUSOjW-A
- Sasig-Roboticoss, E. R. (s/f). Robot Manipulador SCARA con Python y Arduino. Recuperado el 11 de noviembre de 2024, de <https://www.youtube.com/watch?v=r5MFyLbJcDg>

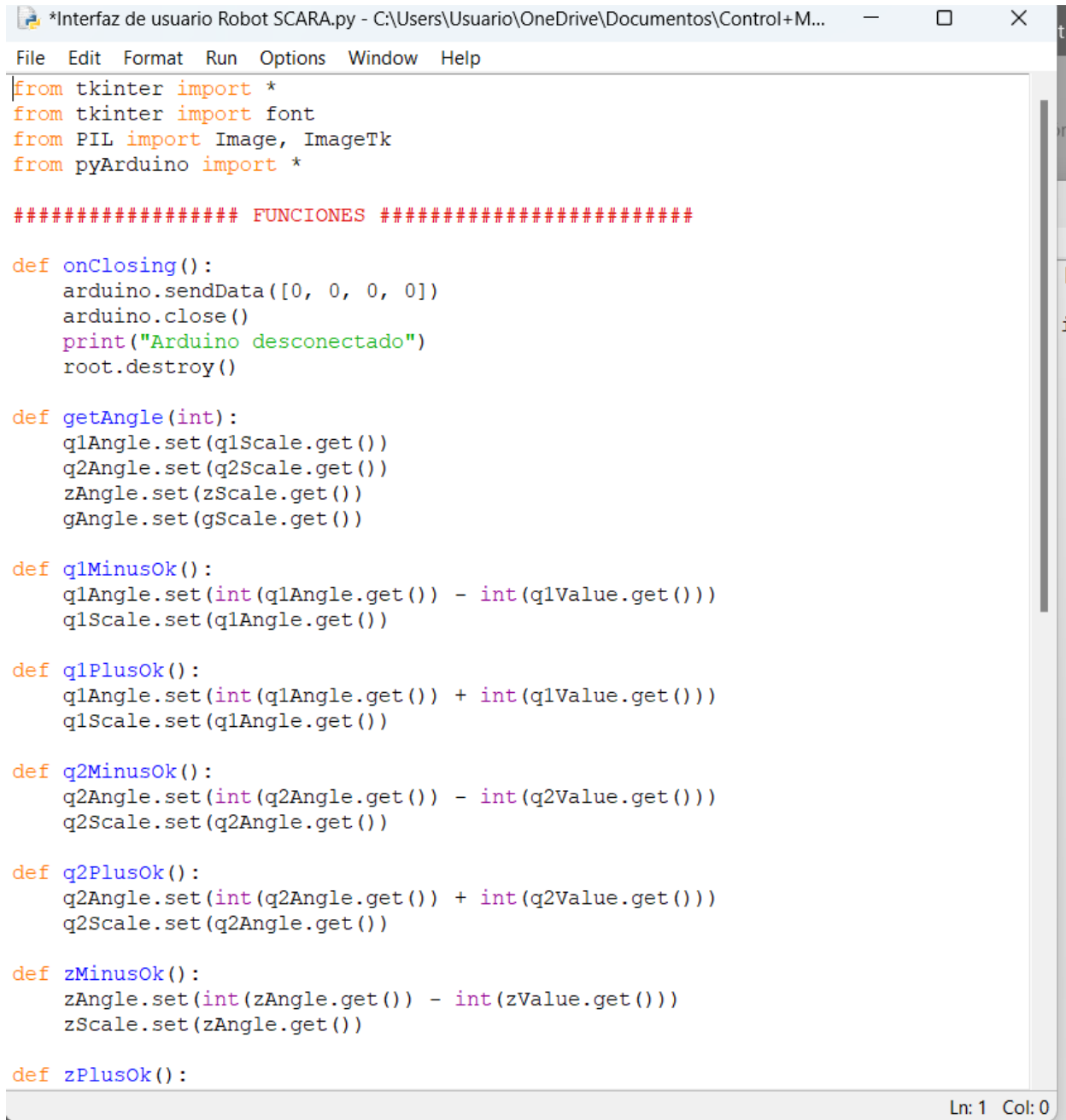
11. Anexos

Anexo A. Funcionamiento Interfaz Gráfica



Disponible en https://drive.google.com/drive/folders/1odt7zYrjUT16B5aDEECpni4_XRCgY2Q?usp=drive_link

Anexo B. Código Interfaz grafica



```

*Interfaz de usuario Robot SCARA.py - C:\Users\Usuario\OneDrive\Documentos\Control+M...
File Edit Format Run Options Window Help
from tkinter import *
from tkinter import font
from PIL import Image, ImageTk
from pyArduino import *

##### FUNCIONES #####

def onClosing():
    arduino.sendData([0, 0, 0, 0])
    arduino.close()
    print("Arduino desconectado")
    root.destroy()

def getAngle(int):
    q1Angle.set(q1Scale.get())
    q2Angle.set(q2Scale.get())
    zAngle.set(zScale.get())
    gAngle.set(gScale.get())

def q1MinusOk():
    q1Angle.set(int(q1Angle.get()) - int(q1Value.get()))
    q1Scale.set(q1Angle.get())

def q1PlusOk():
    q1Angle.set(int(q1Angle.get()) + int(q1Value.get()))
    q1Scale.set(q1Angle.get())

def q2MinusOk():
    q2Angle.set(int(q2Angle.get()) - int(q2Value.get()))
    q2Scale.set(q2Angle.get())

def q2PlusOk():
    q2Angle.set(int(q2Angle.get()) + int(q2Value.get()))
    q2Scale.set(q2Angle.get())

def zMinusOk():
    zAngle.set(int(zAngle.get()) - int(zValue.get()))
    zScale.set(zAngle.get())

def zPlusOk():

```

Ln: 1 Col: 0

Disponible en https://drive.google.com/drive/folders/1odt7z7YrjUT16B5aDEECpni4_XRCgY2Q?usp=sharing

Anexo C. Código Arduino

Archivo Editar Programa Herramientas Ayuda

```

we $
#include <AccelStepper.h>
#include <Servo.h>

////////////////////////////////// ESLABON 1 ////////////////////////////////////
const int J1stepPin = 2;
const int J1dirPin = 5;

AccelStepper J1Motor(1, J1stepPin, J1dirPin); // (Tipo de driver: con 2 pines, STEP, DIR)

// 20 dientes polea motor, 62 dientes polea distal
double q1AngleToSteps = (200.0/360.0)*(62.0/20.0); // (200 pasos/360 grados)x(dientes polea distal / dientes polea motor)
int J1Position = 0;

////////////////////////////////// ESLABON 2 ////////////////////////////////////
const int J2stepPin = 3;
const int J2dirPin = 6;

AccelStepper J2Motor(1, J2stepPin, J2dirPin); // (Tipo de driver: con 2 pines, STEP, DIR)

// 20 dientes polea motor, 62 dientes polea distal , 42 dientes distal
double q2AngleToSteps = (200.0/360.0)*(62.0/20.0); // (200 pasos/360 grados)x(dientes polea distal / dientes polea motor)
int J2Position = 0;

////////////////////////////////// 2 ////////////////////////////////////
const int zstepPin = 4;
const int zdirPin = 7;

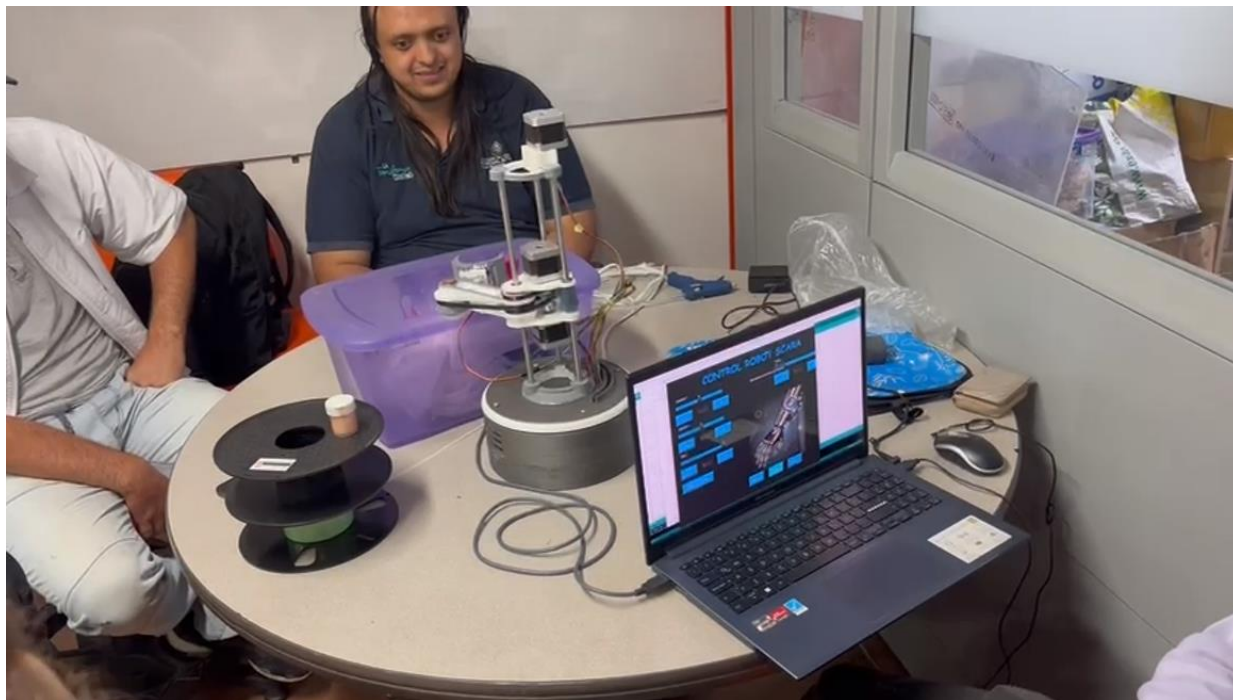
AccelStepper zMotor(1, zstepPin, zdirPin); // (Tipo de driver: con 2 pines, STEP, DIR)
double zDistanceToSteps = 200.0/8.0;
int zPosition = 0;

////////////////////////////////// FINZA ////////////////////////////////////
Servo myServo;
int pinServo = 11;
int angle = 0;
int angleMax = 90;

////////////////////////////////// COMUNICACION SERIAL ////////////////////////////////////
String inputString = "";

```

Disponible en <https://drive.google.com/drive/folders/1-Gix25Z-3uIgEN9H6lXfknr8lHrmVJkd?usp=sharing>

Anexo D. Pruebas internas

Disponible en https://drive.google.com/drive/folders/1Al4Psnfz8EMoyQefyGgcV64HcLaheVb2?usp=drive_link

Anexo F. Fotos y videos de pruebas con estudiantes

Disponible en https://drive.google.com/drive/folders/1A14Psnfz8EMoyQefyGgev64HcLaheVb2?usp=drive_link

Anexo G. Fotos y videos de Ajustes Finales



Disponible en

https://drive.google.com/drive/folders/16CErf2WeCAwJz8Difvt1OcW_d8jQMRTE?usp=drive_link